

Anhang A

Experten-Interviews

A.1 Andreas von Studnitz

Datum: 2. Nov. 2012, Dauer: 1,5 Std.

A.1.1 Kurzfragebogen

Nam : Andr as von Studnitz

FirmaTd[(FiIn)89(tg r_n tz)]TJ0-13.549Td[Pn

v t

o-Erfahrudn

Proz ssmosd ll:sudn27((t scn)89hi dlicn)7(h,s)236(j s)236(nacn)89hnPnartn r-Aog nur;aktu llklassiscnWas-n
d ll.

A.1.2 Transkription

abiaon c

dell.:hez(?n)]TJ/F07 10.9091 Tf 0 -0.3124 Td [(Andreas)-331(v332(on)-331(Studnitr:)]TJ/F17 10.9091 Tf12

festlegenb Daenbanns

te. Inserts,slin

cnhs

201zvn

dafürsm dies saehenballerdingsbbisbjetzts(AnfnNvnimmersoscn

(s)235(eb).rDennoscristzlerKundesnicn(s)235(ernsthaf(s)235(dazus)236(eb))27erti(,s)236(vn)89oorsrGoiar

ensn(s)-989wirkl(cn)89hndassslamitlsosasb

Das zweite ist natürlich, dass man unflexibel wird. Wenn sich Änderungen in den Anforderungen ergeben, muss das über Change Requests gebracht werden, das bringt dann wieder den ganz normalen Ablauf in Gang, d.h. dann müssen wieder erst die Anforderungen analysiert werden, dann muss ein Angebot geschrieben werden, das muss angenommen werden, muss in den Zeitplan integriert werden. Das verzögert den Zeitplan möglichweise und es gibt wenig Möglichkeiten, da wirklich flexibel zu reagieren, ohne alles durchinandern zu bringen. Wir haben noch die Besonderheit, dass wir für inandern Agenturarbeiten, sozusagen Unterauftragnehmern sind und keinen direkten Kontakt zum Kunden haben. Das heißt, es ist immer noch in Ebenen zwischengeschaltet. Wir sind nicht ganz glücklich damit, würden uns eher wünschen, da in flexibleren Modellen zu fahren. Das muss nicht Scrum sein, aber es sollte zum Beispiel Bezahlung nach Aufwand und regelmäßige Überprüfung der aktuellen Anforderungen enthalten. Das wäre schon ein etwas angenehmer und flexiblerer Art zu arbeiten.

FS: Gut, das sind so einige Probleme, die ganz allgemein mit dem klassischen Wasserfallmodell zu tun haben. Gibt es da etwas, was spezifisch für Magento ist oder für Web-Applikationen, was Probleme ergibt oder es noch schwieriger macht? Ist zum Beispiel erhöhte Flexibilität nötig?

AvS: Magento bietet zum Beispiel schon ein Grundgerüst an, in fertigem Layout [Anm.: sieht umso Timplating]. In unserer Angebot steht, dass das Layout nicht zu sehr vom Standard-Layout abweichen darf, zumindest was die Struktur angeht, von in paar vordefinierten Seiten abgesehen. Das Problem ist, Magento bietet 20, 30, 40 verschiedene Seitentypen; wenn man bis ins letzte hinein geht (Wunschliste, Benutzerkonto, Adressverwaltung, Newsletter-Abonnement usw.) und wenn man die alle individuell umsetzen will, dann kann man damit jedes Budget überschreiten. Da wir in festem Budget angeben müssen, andererseits aber nicht das fertige Layout kennen, müssen wir da Bedingung setzen und laufen jetzt im Moment mehrer Schichten, weil inandern Agentur mit der Erstellung des Layouts beauftragt ist, während wir die Timplating-Umsetzung machen. D.h. wir haben das Recht und die Pflicht, über die Layouts der Agentur liefert, drüberzuschauen, zu schauen, ob das dem entspricht was im Angebot steht (ist das mehr, ist da noch etwas unstrukturiert, können wir das so machen, macht das zusätzlichen Aufwand?). Erst wenn wir da alles abgecheckt haben, geht es zum Kunden und der kann das abnehmen. Das ist ziemlich komplizierter Ablauf, das frisst einfach auch Ressourcen. Vorgestern hatten wir zum Beispiel intern Telefonkonferenz mit 3 Leuten unserer Firma, um uns 5 Layouts anzuschauen und zu überprüfen und da haben wir anderthalb Stunden dran gesessen und das ist nicht das erste Mal.

FS: Also verstehe ich das richtig, dass durch die Abhängigkeit von anderen Agenturen bzw. jetzt den Layoutern man erst mitteln im Ablauf konkreter Anforderungen bekommt oder Dinge von denen man abhängig ist und das das ganze Modell kaputt macht?

AvS: Ja. Bezahlungswise macht es das starr und man muss extrem viel Schichten drin um die Layouter dazu zu bekommen, die Seite so zu machen dass sie unskin mehraufwändig verursachen. Das heißt, da haben wir praktisch zwangswise Mehraufwände, die im Ursprungsprojektpreis nicht mitkalkuliert waren.

FS: Würdest du sagen dass es in typischer Konstellation im Magento-Umfeld ist, dass in Firma zum Beispiel sich um das Layout kümmert, und der das Timplating oder vielleicht auch die Modulentwicklung – dass also verschiedene Firmen zusammenarbeiten und von inandern abhängig sind?

AvS: Es ist nicht immer so aber ich habe das schon mehrfach erlebt und das ist in einigen

Proj kt n d r Fall.

FS: Würd st du sag n dass das g n r ll probl matisch ist od r ist das g rad in Sond rfall an d m and r Faktor n b t iligt sind?

AvS: das ist b sond rs dadurch probl matisch dass wir j tzt in n F stpr is hab n. Ich hatt schon in n and r n Fall, da hab ich mit in r Ag ntur in Ki l g arb it t, di das Layout in ng r r Abstimmung mit d m Kund n g macht hab n. Da bin ich nach Aufwand b zahlt word n, das hat dann zwar d n g schätzt n Aufwand d utlich übrschritt n w il di Anford rung n Schritt für Schritt imm r w it r g sti g n sind, doch s war in w s ntlich ang n hm r s Zusamm narb it n und das Erg bnis war dann auch l tztlich s hr s hr gut. Ab r da muss d r Kund auch ntspr ch nd fl xib l s in und g g b n nfalls akz pti r n, dass r für rhöht Anford rung n auch m hr G ld b zahl n muss und s twas m hr Z it braucht. Grundsätzlich ist s so, ich arb it imm r w s ntlich li b r w nn ich nach Aufwand arb it n kann, da ist j tzt auch ab Januar in n u s Proj kt für in n Kund n in d r Schw iz in Aussicht, für d n ich schon m hrfach g arb it t hab . Das war imm r in ang n hm s zusamm narb it n auch mit s hr gut n R sultat n.

FS: D r Unt rschi d zwisch n F stpr is und B zahlung nach Aufwand hängt ja auch s hr davon ab, wi gut di Aufwandsschätzung ist. Vi ll icht unabhängig davon, ob zwisch n durch n u Anford rung n hinzukomm n: wi gut funktioni r n di Aufwandsschätzung n b i Mag nto-Proj kt n g n r ll?

AvS: mit d r Erfahrung wird s d utlich l icht r. In d n rst n in, zw i Jahr n hatt ich öft r das Probl m, dass m in Aufwandsschätzung d utlich zu g ring war w il ich infach manch Mag nto-Probl m nicht so inschätz n konnt und nicht im Vorhin in g s h n hab ab r s it ich zw i, dr i Jahr Mag nto-Erfahrung hab und das Syst m ntspr ch nd auch von inn n k nn , w iß ich im Prinzip schon vorh r, wi ich in Anford rung ums tz n würd , wo Probl m zu rwart n sind, wo k in Probl m zu rwart n sind und wi d r Aufwand ung fähr ist. Es kann natürlich s in, dass ich mal um in, zw i Tag dan b n li g ab r s gibt k in grob n Schnitz r. In d n l tzt n Monat n bis in, zw i Jahr n hab ich ig ntlich imm r m in Schätzung n ing halt n sof rn k in n u n Anford rung n g komm n sind.

FS: Man muss dazu ab r schon das Syst m „von inn n k nn n“, wi du sagst, also nicht nur damit g arb it t/ ntwick lt hab n, sond rn schon auch wiss n, wi s int rn funktioni rt

AvS: s hilft natürlich auch s hr, w nn man ähnlich Anford rung n schon mal in and r n Proj kt n umg s tzt hat. Ansonst n hab ich di Erfahrung g macht, dass vi l di noch nicht so lang mit Mag nto g arb it t hab n, das Syst m und di Aufwänd di dann anfal l n unt rschätz n. Das Syst m ist xtr m fl xib l ab r s hat auch d n Nacht il, dass di s Fl xibilität w nn man si d nn b wahr n will auch rhöht Anford rung n an d n Entwickl r st ll.

FS: Es ist nach m in n Erfahrung n und d m was ich bish r g hört hab nicht ganz infach, so in ti f s Wiss n üb r di int rn n Abläuf zu rlang n. Was sind d nn da so di Hind rniss und Schwi rigk it n?

AvS: zum in n dass s grundsätzlich rst mal unt rschätzt wird. Es gibt vi l Entwickl r – zu d n n ich auch g hör – di h r aus d r PHP-Schi n komm n und vorh r and r PHP-Proj kt b arb it t hab n. B i mir war n s Typo3 und xt:Comm re , di w s ntlich infach r w it rzu ntwick ln sind. D m ntspr ch nd wird Mag nto s hr häufig unt rschätzt und di Anford rung n di s an di Einarb itung st ll. Es ist infach in s hr kompl x s

Syst m mit inig n Anl ih n aus d r Softwar t chnik, di s in and r n PHP-Proj kt n so nicht gibt, und di schon Anford rung n an d n Entwickl r st ll n. Was da hilft, ist sich rlich in akad misch Ausbildung in d r Informatik, das hat mir zumind st g holf n. Was natürl ich auch hilft ist allg m in PHP-Erfahrung b sond rs im obj ktori nti rt n B r ich, was hilft ist in gut s Buch dazu zu l s n und natürl ich in gut Schulung zu b such n, di s ab r zu m in n Anfangsz it n noch nicht gab. Das h ißt, dadurch hat für mich zum B ispi l d r Einsti g d utlich läng r g dau rt als s j tzt für j mand n d r mit m in m Hint rgrund anfang dau rn würd , d r sich in Schulung l ist n könn t bzw. b zahlt b käm . Das hilft auf j d n Fall schon mal inig s. Ein n Anspr chpartn r zu hab n, d m man Frag n st ll n kann ist auch vi l, vi l w rt. Ich hab das b i Kund n von mir rl bt, da hab ich in n Entwickl r für zw i Tag g schult und war dann imm r wi d r im Proj kt b gl it nd dab i, r hat mir b i Skyp dann imm r Frag n st ll n könn n und d r ist xtr m schn ll r ing komm n. Er ist j tzt in dr ivi rt l Jahr dab i, r hat noch nicht m in Niv au ab r r kann schon s hr vi l s lbst.

FS: Ohn Schulung od r konkr t n Anspr chpartn r sind di Möglichk it n ja r cht g ring. Es gibt in gut s Buch, das d ckt ab r nun auch nicht all s ab, s gibt k in offizi ll Doku m ntation, d r Cod s lbst ist w nig dokum nti rt...

AvS: w nn man sich das s lb r b ibring n will, muss man rst mal d n Einsti g irg ndwi g schafft hab n. Erst mal in, zw i Modul mit Hilf aus d m Int rn t g bast lt hab n, vi ll icht mal in K rnmodul hin ing guckt. J m hr man da si ht und v rst ht und rk nnt, d sto m hr v rst ht man das ganz Mag nto-Syst m, und rst dann kann man auch v rnünftig nach d n gl ich n Standards gut Mag nto-Modul ntwick ln

FS: Wo du g rad Standards und gut Modul r wähnst... Mag nto rmöglich t ja zi mlich vi l Fl xibilität. Manch r würd sag n zu vi l, da s vi l v rschi d n Möglichk it n gibt, in Zi l zu rr ich n und di für Anfäng r vi ll icht am infachst n auss h nd ist, infach Klass n zu üb rschr ib n.

AvS: ich v rgl ich s imm r g rn mit Entwicklung für Typo3. dort gab s vi l m hr Möglichk it n, das fand ich h r noch schl cht r. B i Mag nto s h ich s ig ntlich so: s gibt häufig b i n u n Anford rung n nur in od r zw i W g wi man so was v rnünftig mach n kann. W nn man di ntspr ch nd Erfahrung hat und s inig rmaß n standardkonform und upgrad fähig ntwick ln will, dann gibt s nicht so vi l Möglichk it n, das sinnvoll zu mach n. Darin s h ich ig ntlich auch in n Vort il von Mag nto w shalb man auch fr md Modul s hr schn ll üb rblick n kann und s hr schn ll si ht, was si mach n. Man schaut sich di Konfigurationsdat i an und w iß ig ntlich schon in 2 Minut n, was di s s Modul üb rhaupt macht: was macht s mit d m K rnsyst m, wo bringt s n u V rknüpfungspunkt zum K rnsyst m an, w lch Klass n üb rschr ibt s, wo s tzt s Obs rv r in, w lch n u n Controll r d fini rt s und so w it r.

FS: ... w nn das all s standardkonform g macht ist sagst du. Sind d nn di g g b n n Standards ausr ich nd, ausr ich nd dokum nti rt, brauchbar?

AvS: Dokum nti rt ist das nicht v rnünftig. Am b st n kann man sich di Standards in d n K rnmodul n ans h n; man hat d n Standard dann rr icht, w nn das ig n Modul ähnlich auss ht wi in K rnmodul. Es gibt da auch infach r ab r das sind ganz gut B ispi l , nach d n n man sich richt n sollt und di auch allg m in als B st Practic ang s h n w rd n.

FS: Man muss sich das also schon irg ndwi aus B st h nd m h rl it n.

AvS: Es ist sowi so in s hr v rbr it t Praxis, s hr vi l Copy & Past inzus tz n. Nach- s h n, w lch s Modul in ähnlich Funktionalität hat (zum B ispi l n u r Tab im B nutz r- konto: Wunschlist nmodul), und schau n was g nau das macht. Man kopi rt sich di ntspr - ch nd n Til , passt di Id ntifi r an und s tzt s in n ig n n Cod dazu. Man muss dazu nur wiss n, wo s j w ils st ht.

FS: Du hast vorhin rzählt, dass man sich b im Einsatz von Fr mdmodul n zunächst di Konfiguration anschaut, schaut was gibt s da für R writ s und so w it r. Wi st ht s d nn g n r ll mit d r Int gration von v rschi d n n Modul n aus v rschi d n n Qu ll n od r n u n Modul n in b st h nd n Shops: gibt s dab i Probl m ?

AvS: Ja, da gibt s Probl m . Ich k nn inig L ut , di sag n, am b st n so w nig Fr mdmo- dul wi möglich. And r s tz n r lativ b d nk nlos Fr mdmodul in, ab r m in M inung ist auch h r, w nn man auf in Fr mdmodul v rzicht n kann, sollt man das tun. So hat man s lbst di voll Kontroll , w il man b n doch nicht imm r od r zumind st nicht ohn Aufwand 100% w iß, was in Fr mdmodul macht, ob nicht doch irg ndwo in Konflikt ist. W nn man s tut, muss man natürlich auf Konflikt acht n. Es gibt da schön Tools, um di R writ s zu üb rwach n, dass nicht zw i Modul das gl ich Mod l od r d n gl ich n Block üb rschr ib n ab r auch dafür dass man di s Möglichk it üb rhaupt in B tracht zi ht ist Erfahrung notw ndig. Sonst st ht man irg ndwann vor in m F hl r, d r auf d n rst n Blick nicht rklärbar ist.

FS: Hat das nur mit unt rschi dlich r Qualität zu tun und dass man nicht sich r s in kann, ob das Modul j tzt standardkonform ist, od r ist das in g n r ll s Probl m von Mag nto?

AvS: Das ist h r in g n r ll s Probl m. Es gibt Art n von Modul n, di ich r lativ b d nk nlos ins tz . Dazu g hör n Zahlungs- od r V rsandart n, w il di üblich rw is k in od r w nig Mod ls und Blöck üb rschr ib n, sond rn an di ntspr ch nd Standardschnittst ll von Mag nto andock n.

FS: Also hat man in d m Fall in Schnittst ll , di s hr klar d fini rt ist, und mit d r man nicht vi l kaputt mach n kann...

AvS: Ja, w nn man nicht zusätzlich noch di K rnfunktionalität rw it rn will. Zu V rsandart n zum B ispi l kann man noch in M ng dazu ntwick ln an Zusatzfunktion n. Will ich zum B ispi l noch in H b bühn dazu b st ll n, muss ich doch irg ndwann an di K rn- Mod ls h ran und w nn dann zw i Modul das gl ich mach n, b kommt man doch wi d r Probl m . Ab r g n r ll ist zu sag n, Zahlungs und V rsandart n sind h r unkritisch. Was vi l kritisch r ist, sind Modul , di b st h nd Funktionalität anpass n. Da kann di Cod - Qualität noch so gut s in, s ist häufig rford rlich dass man Klass n üb rschr ibt und das si ht man dann ntspr ch nd auch b i Modul n von wirklich gut n Entwickl rn. Vinai Kopp zum B ispi l war von Anfang an dab i, li f rt xtr m hoh Qualiät ab und arb it t mitt- l rw il b i Mag nto. Ab r auch r hat Modul h rausg bracht, di nicht unt r 5 R writ s auskomm n und w nn man so twas ins tzt, muss man imm r aufpass n.

FS: Nun hat Mag nto an inig n St ll n wi V rsandmodul n also r cht klar Hot Spots, in di man sich s in Funktionalität inbaut ohn Konflikt zu b komm n, hat w it rhin s hr vi l Ev nts, wo man sich inhäng n kann. Wi kommt s, dass trotz d m so vi l R writ s nötig sind?

AvS: Es sind nicht g nug Ev nts. Es f hl n an z ntral n St ll n imm r noch Ev nts, in klassisch s B ispi l ist di Anpassung d r B st llb stätigungs-Email. J d s mal muss man

das komplette Ordner-Modell überschreiben, wenn man irgendwas daran anpassen will. Da war die Firma Magento auch nicht offengemutig, neue Events einzuführen. Ich hoffe, mit Magento 2 wird das besser; dass sich da im Vorfeld auch gemutigter entwickeln dranbelassen und andernfalls den Stellen Patch committen, um selbst Events einzubringen. Es wird nicht der Fall sein, dass allen notwendigen Stellen Events sind, aber es könnte noch deutlich besser sein.

FS: Es fehlt also in gewisser Konsequenz in Magento selbst?

AvS: Ja. Also schließlich an vielen Stellen; einige Frontendmodule nutzen das Event-System nicht, auch wenn es möglich ist, weil es tiefgehendes Wissen erfordert. Zum Beispiel die Anpassung von verschiedenen Tabellen und Formularen im Backend: das geht üblicherweise ohne Rewrite, nur mit Event/Observer aber viele wissen oder kennen es halt nicht, ich hab selbst erst relativ spät herausgefunden, wie das geht. Dementsprechend haben viele Module noch unnötig viele Rewrite drin.

FS: Das lässt sich wohl wieder auf nicht existente Dokumentation zurückführen. Man muss sich alles selbst zusammensuchen?

AvS: Genau. Mittlerweile gibt es sehr gute Ressourcen im Web aber das ist auch Stückwerk, nicht an der Stelle zusammengeschrieben. In diesem Blog in guter Bitrage dazu, in jenem in guter dazu, teilweise auch etwas im offiziellen Magento Wiki oder in dem Magento-Forum, aber es gibt halt kein zentrales Stellen dafür. Aber so in Entwicklungsdokumentation müsste auch über 1000 Seiten stark sein, würde ich vermuten. Die würde sich dann wahrscheinlich in der mehr durchlesen.

FS: Sie wäre ja nicht nur zum durchlesen, sondern auch zum gezielten Nachschlag hilfreich...

AvS: Die Grundlagen muss man sich auf jeden Fall erst mal aneignen bevor man in die Details gehen kann. Als Einstiegspunkt würde ich immer in Schulung von zumindest einm Tag empfehlen. Dann kommt man deutlich schneller weiter, dann findet man auch die richtigen Stichpunkte zum Suchen, weißt die Kernmodule aufgebaut sind und wo man nachsehen kann wenn man in bestimmmte Funktionalität sucht und wo sie umgesetzt wird.

FS: Gehn wir mal in den Schritt zurück von Implementierung und Integration zum Entwurf. Bevor man einfach anfängt draufloszuschreiben. Du sagst, selbst als erfahrener Entwickler weißt du oft noch nicht so genau, was gibt es zum Beispiel für Events die man nutzen könnte, müsstest erst mal suchen wie man dies und jenes umsetzen kann, wenn du es nicht schon mal gemacht hat. Stellt das beim Entwurf ein Problem dar?

AvS: Es ist natürlich ein Problem, wenn man beim Entwurf noch nicht genau weiß wie man es umsetzen kann. Es besteht immer erst in Rechen-Schritt.

FS: Erzähl doch mal, wie das bei euch abläuft wenn es jetzt um die Modulentwicklung geht. Beim Wasserfall wäre es ja theoretisch so, dass man den kompletten Entwurf fertigstellt, bevor man anfängt zu codieren.

AvS: Man bekommt zumindest erst mal in Anforderungsdokument. Wenn ich auf der Basis in den Preis abgeben soll, mache ich mir in dem Moment schon mal Gedanken, wie ich es umsetzen will. Da hilft es mir sehr, die einzelnen Teilanforderungen runterschreiben. Wenn ich das entsprechend runtergebrochen hab, dann kann ich auch jeweils zu den einzelnen Punkten in den Preis begeben, weil ich mir dann im Kopf schon überlegen kann, wie ich es

ums tz . T ilw is muss ich zu d m Zitpunkt ab r auch schon in rst R ch rech -Rund
inl g n und guck n, wo gibt s pass nd Ev nts, w leh Klass n muss ich anpass n usw.

FS: Du hast also ig ntlich in n Schritt zu früh schon s hr vi l mit D tails zu tun?

AvS: Ja, schon. Ich muss auch t ilw is in d m Schritt schon inig Sach n ausprobi r n.
Da muss man noch k in ig n s Modul schr ib n, das g ht auch mal schn ll im Cor , w nn
man s danach wi d r rückgängig macht. Ab r dann w iß man w nigst ns wo in Änd rung
ang s tzt w rd n kann und hat sozusag n schon s in Ums tzungssp zifikation f rtig. Das ist
natürlich twas zu vi l Arb it zu d m Zitpunkt.

FS: Also ist in Probl m b i d r Sp zifikation, dass man schon in, zw i Schritt vorgr if n
muss...

AvS: Od r man hat s halt schon mal ähnlich in and r n Proj kt n g macht und w iß g nau,
wi man vorg ht.

FS: Gibt s noch and r Probl m , di b i d r Sp zifikation auftr t n?

AvS: Di Anford rung n sind s hr häufig nicht sp zifisch g nug, nicht durchdacht g nug,
vi l D tailfrag n müss n noch g klärt w rd n, di t ilw is in n größ r n Einfluss auf d n
Aufwand hab n. Ich rl b s imm r wi d r, dass ich in in m Ang bot bzw. in m in m
Sp zifikationsdokum nt Annahm n tr ff n muss od r noch in, zw i Rückfrag rund n mit
d m Kund n v r inbar n muss. Ab r das ist glaub ich in allg m in s Softwar probl m.

FS: Ist s d nn in sp zi ll s Probl m von Anw ndungsfram works, dass Kund n oft zu w nig
K nntnis von d r Grundanw ndung hab n und davon, w leh F atur s schon nthalt n sind?

AvS: Ja, t ilw is sich rlich. Das kommt imm r auf d n Kund n an. Es gibt Kund n mit d n n
kann man in d m Sinn s hr ang n hm arb it n w il si sich schon mit Mag nto b schäftigt
hab n und d m ntspr ch nd nur das b schr ib n was vom Mag nto Standard abw icht, und
and r Kund n, di Schritt für Schritt b schr ib n, was in Shop könn n muss. Das muss man
dann s lbst rst mal mit d n Sp zifikation n von Mag nto abgl ich n und gibt s t ilw is
ganz infach Sach n, di ig ntlich zi mlich unwichtig sind, di ab r Mag nto im Standard
infach and rs macht und di dann m hr r Tag Aufwand v ursach n würd n, w nn man s
so mach n würd wi im Anford rungsdokum nt. Und das ist dann häufig nicht ganz infach,
d m Kund n auszur d n.

Aus d m s lb n Grund ist s auch für d n Entwickl r wichtig, das ganz Mag nto Fram work
zu k nn n, zu wiss n was sind di inz ln n F atur s, und imm r auf d m Lauf nd n zu bl i-
b n. Damit man nicht in Anford rung b r chn t di d r Mag nto-Standard längst kann und
dann ggf. n u ntwick lt. Das ist für mich zi mlich wichtig, sich imm r auf d m Lauf nd n zu
halt n was di aktu ll Mag nto-Entwicklung ang ht, nicht nur das was im K rn passi rt, son-
d rn auch was in d r sog nannt n Community passi rt. Es gibt da ja schon inig int r ssant
Modul und Erl icht rung n, Entwicklungshilf n und ähnlich s, di in n w it r bring n. Als
B ispi l für di Entwicklungsumg bung PHPStorm das r lativ n u Plugin Magic nto, das
di Entwicklung d utlich v r infacht

FS: Was macht das g nau?

AvS: B ispi lsw is bi t t s b i Variabl n, di aus in r Factory b füllt w rd n, di
Möglichk it, dir kt in di richtig Klass hin inzuspring n und nicht nach cor /abstract
[M ge_Core_Model_Abstr ct], od r automatisch ntspr ch nd @v r Annotations für and -

r IDEs inzuführen. Wichtig kann es automatisch Rewrite für passende Klassen anlegen und ige Tmplat -Kopien im richtigen Thema -Verzeichnis anlegen. Und es enthält einige Shortcuts, zum Beispiel um die config.xml schnell zu erstellen. Dort bietet/schlägt es die möglichen XML-Tags vor. Da kommen einige nette Sachen zusammen, die bisher noch nicht gab.

FS: Das adressiert ja auch wieder einige Probleme, die einfach aus der Magento-Architektur herrühren und wo das Plugin noch nicht nutzt, hat dies Schwirigkeiten möglich gemacht.

AvS: Richtig.

FS: Was sind denn das genau für Probleme, zum Beispiel in Bezug auf die Factory-Methoden?

AvS: [erklärt Service Locator, Rewrite] Wenn man sich mit dem System nicht auskennt, ist es schwierig herauszufinden, welche Klasse catalog/product überhaupt ist und wo finde ich die zugehörigen Methoden des vorliegenden Objekts. Da gibt es natürlich Möglichkeiten, die der Entwickler aber erstmal kennen muss, und Umwege die er gehen muss, denn die Unterstützung in der Entwicklungsumgebung fehlt bisher. Dies ist durch Magento zumindest deutlich besser gegeben, wenn auch noch nicht perfekt. Genau wie die XML-Daten, die ja in Schema-Definition haben weil sie sich für die Aufbauarbeiten eignen, sind das Probleme die vor allem Anfänger vor großen Hürden stellen, egal ob man viel Programmiererfahrung mit anderen Sprachen oder Systemen hat. Mit der Erfahrung wird dies Problem geringer.

Das Templating-System von Magento ist auch nicht ganz einfach, weil es durch die Layout-XMLs gesteuert wird. Da habe ich auch erst sehr spät herausgefunden, wie das jetzt richtig funktioniert, wie man ohne Frickel in Änderungen machen kann, wie man es wirklich sauber macht, so dass man auch einige Maßnahmen updaten kann. Das heißt, dass man nicht das ganze Tmplat überschreibt oder sogar das Kern-Tmplat anpasst, sondern nur die nötigen Tmplat -Daten in seinem Themaverzeichnis packt und dass man analog auch nicht komplette Layout-XML-Daten überschreibt, sondern nur die Änderungen in einigen Daten packt, beispielsweise die loc 1.xml.

FS: ...und wenn man das nicht ganz sauber macht, werden Tmplat überschrieben.

AvS: ...und die Update-Sicherheit ist nicht mehr gegeben. Damit kann es schon sehr schwer sein, in Magento-System zum Beispiel von 1.3 auf 1.7 upzudaten.

FS: Das ist wieder ein typisches Problem bei der Einbindung von externen Modulen, richtig?

AvS: Ja, wobei externe Module mit Tmplat Erfahrungsgemäß weniger Probleme machen, die bringen ihre eigenen Tmplat mit und überschreiben nichts. Aber ich habe mehrer System gesehen, die von Agenturen oder freien Entwicklern umgesetzt worden sind und wo dann wirklich das ganze Tmplatverzeichnis übernommen wurde. Das macht in Update schon sehr schwierig. Das hat wenig mit den Extensions zu tun, sondern mehr mit demjenigen, der für die Umsetzung des Magento-Projekts ursprünglich zuständig war.

FS: Trotzdem nicht schlecht...

AvS: Das ist so ein typisches Anforderung an Magento-Entwickler: „Mein Entwickler hat mich verlassen oder kommt nicht mehr wieder oder ich bin unzufrieden mit ihm, können sie nicht mal die Sache Projekt übernehmen?“ – das ist ein undankbare Aufgabe, weil da kein Shopbetreiber

gern hören möchte, dass fünfstellig B-träger notwendig sind um das wiedergrad zuzuhören. Aber häufig ist das halt der Fall.

FS: Bei Updates und Änderung von überronnenen Shops, will man ja sicherstellen dass bereits vorhandene Funktionalität bestehen bleibt. Hier kommen Tests ins Spiel; willst du dich Tests umsetzen und was gibt es dabei für Probleme?

AvS: Bisher habe ich hauptsächlich manuelle Tests und das klappt nur bedingt weil in der Person oder auch in der gesamten Funktion kontinuierlich durchtesten kann. Wenn man wirklich auf Nummer sicher gehen will, muss man automatisch Tests irgend in der Art einsetzen, ansonsten wird früher oder später der Kunde noch unentdeckte Fehler finden. Zum Beispiel haben wir kürzlich in anderen Projekten von anderen Agenturen übernommen, von 1.3 auf 1.7 hochgezogen und Monat später wurde festgestellt, dass die Vergleichsfunktion überhaupt nicht mehr funktioniert, weil in der Testumgebung nicht mehr vorhanden war. Klar, da wünscht man sich automatisierte Tests, aber es gibt sie halt nicht und sie zu erstellen ist großer Aufwand, den kaum ein Kunde bezahlen möchte.

FS: Es gibt nun seit einiger Zeit das Magento Test Automation Framework für Unternehmen. Zwischendurch, hast du damit Erfahrung?

AvS: Da habe ich noch nicht mitgearbeitet, soweit ich weiß, testet es aber hauptsächlich das Backend, ist das richtig?

FS: Es sollte eigentlich alle Kernfunktionen mit Selenium-Tests abdecken, also neben dem Backend auch die Standardabläufe bei Warenkorb, Checkout und so weiter...

AvS: Das wäre dann sicherlich in der Möglichkeit, die Standardfunktionen zu testen. Es bedingt natürlich, dass kein relevanten Änderungen an den Standardfunktionen vorgenommen wurden. Das heißt, im Idealfall setzt man dieses Framework von Anfang an ein, lässt es die Nacht hindurch laufen und passt es Stück für Stück an die jeweiligen Anpassungen an. Es hat allerdings immer den Nachteil, dass in der Testumgebung der Stunden dauert. Ich würde daher normalerweise dazu überlegen, dass nur die systemkritischen Funktionen getestet werden, wie Checkout und Anmeldefunktion und dazu die Sachen, die man selbst anpasst. Aber das ist eher Theorie, etwas was ich gerne machen würde, wo ich aber bisher nicht weiterdringlich nicht hatte oder das nötig Budget fehlte.

FS: Kommt es deiner Erfahrung nach denn häufig vor, dass Modul X in Modul Y oder dem Magento-Kern unentdeckte Seitenfehler hat, die man wahrscheinlich erst mit einem automatischem Test unmittelbar gefunden hätte?

AvS: Ich weiß gerade kein konkretes Beispiel aber es ist tatsächlich schon öfter vorgekommen, dass in der Änderung Einfluss auf etwas hatte, womit es eigentlich nichts direkt zu tun hatte. Besonders bei komplexen Änderungen zum Beispiel am Kundenmodell hat man eventuell nicht immer alle Spezialfälle im Überblick. Wenn man dann irgendwann mal die Standardfunktion nutzt, die bei der Entwicklung von Zusatzfunktionen nicht berücksichtigt wurden (zum Beispiel die Flat Product Tables), dann können Bugs anstellen auftreten, an denen man sich nicht vermutet hätte. Das ist mir durchaus schon passiert; das wäre natürlich schon gewiss, angesprochen Tests zu haben, die müssen aber natürlich jeweils mit den notwendigen Änderungen und Einstellungen gepflegt werden. Solche Selenium-Tests kann man auch eigentlich nur am Live-System sinnvoll anwenden, sodass denn man hat in der Testumgebung, bei der es sichergestellt ist, dass es genau auf dem gleichen Stand ist.

FS: Ext rn Einflüss wi di Syst mkonfiguration hab n also auch noch Auswirkung n...

AvS: Genau. Wir arb it n aktu ll an in m Proj kt mit m hr r n Entwickl rn, wo wir auch nicht di inzig Firma, sind di da mit ntwick lt. Da sind wir dazu üb rg gang n, sämtlich Konfigurations-Änd rung n nicht m hr dir kt in d r Dat nbank, also üb r di Anw ndung, vorzun hm n, sond rn üb r in Updat -Skript im Modul. Das rford rt rst mal m hr Aufwand, sorgt ab r dafür, dass sämtlich Instanz n auf d m s lb n Stand sind: Kat gori n, CMS-S it n tc. – all s üb r PHP bzw. SQL Cod . Das ist schon in qualitätssich rnd Maßnahm , di sich rst llt dass all Instanz n b i all n 5 Entwickl rn bis hin zu T st-, Staging- und Liv -Syst m auf d m s lb n Stand sind.

FS: Das b trifft j tzt di Mag nto-Konfiguration. Nun ist s g rad b i PHP ja so, dass auch noch vi l von d r Syst mkonfiguration abhängig ist, wi d r global n `php.ini` ist das in Probl m od r wird das vom Fram work ausr ich nd abstrahi rt?

AvS: Ich hab von vi l n Lut n g hört, dass si das auch abstrahi r n, ind m si auf all n Entwicklungsinstanz n di gl ich Umg bung ins tz n. Ich mach das nicht, hab damit ab r auch w nig schl cht Erfahrung n g macht. Es ist zwar schon mal passi rt, dass s Groß-/Kl inschr ibungsprobl m gab w il ich Windows ins tz – Windows ist fl xib l was Groß-/Kl inschr ibung ang ht – und da war dann in Dat inam groß statt kl in g schri b n und Linux auf d m S rv r hat di Dat i nicht g fund n, ab r so twas passi rt m in r Erfahrung nach r lativ s lt n. Es kann natürlich passi r n, und s kann auch and r S it n ff kt durch di Syst mkonfiguration g b n, zum B ispi l b i unzur ich nd m Sp ich rlimit. M in r M inung nach ist so in synchronisi rt Umg bung ab r nur b i größ r n Proj kt n sinnvoll od r w nn b r its in vorb r it t Umg bung xisti rt, di für all Proj kt v rw nd t wird.

FS: J tzt hast du, wi ich w iß, mittl rw il auch Erfahrung mit Unit T sts. Wi si ht s damit aus, gibt s da b züglich Mag nto Probl m ?

AvS: Für das aktu ll Proj kt gab s di inform ll Vorgab , di Til , di n u ntwick lt w rd n, sollt n kompl tt mit Unit T sts abg d ckt s in; di and r Ag ntur mach das mit all n Proj kt n. Wir hab n d m mündlich zug stimmt, d m ntspr ch nd hab ich mich in das Th ma ing arb it t und s für di rst n Modul auch g macht. M in Erfahrung damit war all rdings bish r nicht so gut, was glaub ich hauptsächlich damit zusamm nhängt, dass ich mit d n T sts noch nicht so gut ausk nn und ni mand n hatt d n ich da anspr ch n konnt , s w nig Dokum ntation gibt. Grundsätzlich gibt s ja zw i T st-Fram works di auf PHPUnit basi r n, da hab ich das von Ivan Ch purnyi (EcomD v) ing s tzt, was glaub ich auch das umfangr ich r ist und m hr Möglichk it n bi t t. Es rmöglicht zum B ispi l, Produkt zu muli r n und all möglich n and r n Mod ls. Ab r irg ndwann bin ich dann doch g sch it rt w il ich b i Sp zialfäll n nicht m hr wusst , wi ich das v rnünftig ins tz n kann und infach nicht m hr w it rg komm n bin. Ich hab auch das Probl m, dass ich s hr rfahr n mit Mag nto bin ab r s hr w nig rfahr n mit Unit T sts. Das h ißt, w nn in T st fhlschlug, lag das m ist ns am T st, nicht am Produktionscod . Das hat nur zusätzlich Unsich rh it hin ing bracht, ich bin für mich j tzt vor rst zu d m Schluss g komm n, dass s mir nichts bringt, abg s h n von absolut n „Basic-T sts“ wi : xisti rt in Layout-Dat i, ist in Obs rv r korr kt d fini rt, so twas. Ich w rd v rmutlich im Januar an in m zw itägig n Workshop von Ivan Ch purnyi t iln hm n und hoff , dadurch in n fundi rt r n Einsti g zu b komm n. Grundsätzlich d nk ich, dass s sinnvoll ist, Unit T sts inzus tz n, zumind st bis zu in r g wiss n Eb n , wo man dann auf funktional T sts umschw nk n sollt

FS: War s h r in Probl m, wi di T sts üb rhaupt aufg baut sind, od r ist s h r das

kompliziert Fixtur -System?

AvS: Ehrlicherweise: wie kriegt ich die richtigen Daten in die Tests, dass die Tests auch tatsächlich funktionieren, wie es geplant war. Wie bekomme ich über die Fixtur in korrekter konfigurierter Produkt, genau so wie ich es im Backend anlegen würde oder importieren würde.

FS: Meinst du, dass das hier in Problem des Magento-Datenmodells ist, oder in Problem der Fixtur -Implementation des Test-Frameworks?

AvS: Für mich ist es in Problem der Dokumentation. Es gibt nicht für alle Modelle, die man gerne nutzen würde, Beispiele. Für die einfachen kann ich das noch selbst herleiten, aber sobald es komplexer wird, mit Abhängigkeiten zwischen den Modulen wie zum Beispiel Relations, Bundles, Products oder ähnliches, dann brauche ich Beispiele um damit vernünftig zu arbeiten. Und da ist die Dokumentation doch zu dünn. Wie die Daten in der Datenbank aussahen würde, weiß ich, aber nicht wie ich die entsprechenden Fixtures schreiben muss. Und dazu kommt, dass ich die Abläufe des Magento PHPUnit Moduls nicht ausrechnen kann.

FS: Du sagst, du kommst aus der PHP-Ecke und hast auch Erfahrung mit anderen PHP-Projekten zum Vergleich. Von dem Problem über die wir bisher gesprochen haben: kannst du vielleicht identifizieren, die in anderen Problemen von PHP-Frameworks sind, und nicht Magento-spezifisch? Was lässt sich zum Beispiel auf TYPO3 übertragen?

AvS: Bei TYPO3 hatte ich grundsätzlich das Problem, dass der Compiler relativ abgeschottet war, dass die Kernelmodule häufig in ihrer eigenen Technik entwickelt wurden; zumindest hatte ich diesen Eindruck. Das finde ich bei Magento so angehen, dass die eigenen Module so aussahen wie die Kernelmodule. Bei TYPO3 hat man also deutlich weniger gute Beispiele und Vorgaben. Ext:Commerce ist sowieso etwas anders, weil es sehr „gskript“ ist und man da eigentlich sowieso nur Anpassungen am Kernsystem macht, nicht eigenständiges Modul.

FS: Das sind also drei Anwendungen mit sehr unterschiedlicher Architektur...

AvS: Magento bietet darunter mit Abstand die besten Grundlagen dafür, hochqualitativ zu entwickeln. Die Ähnlichkeit zwischen Kernelmodulen und eigenen Modulen ist insbesondere wichtiger Faktor, also dass man einfach sehen kann, wie ein Modul „gut“ entwickelt ist. Das war bei Ext:Commerce überhaupt nicht so und bei TYPO3 nur sehr eingeschränkt.

FS: Sprich, Entwurf und Implementierung stellen sich sehr unterschiedlich dar aufgrund der verschiedenen Architekturen. Wie ist es mit der Anforderungsanalyse und Spezifikation, lassen sich da die Probleme bei Magento-Projekten auf andere Frameworks übertragen?

AvS: Auf Ext:Commerce eigentlich nicht, weil man sich da zum Beispiel sehr wenig Gedanken um Update-Sicherheit machen muss, weil es sowieso kein gab. Immer wenn man da etwas programmiert hat, war das im Kernsystem, damit konnte man dann je nach Updateversion. Man konnte immer noch externen Modul einbinden aber das lief dann immer so ab, dass es in der Beschreibung dazu gibt, in der steht: „In die Datenbank die Zilbitte die Zilbitte einfügen“, in paarigen Datenவில் nicht noch, aber im großen und ganzen wurde das System dann zusammengestellt. Das ist mit TYPO3 oder Magento dann doch ganz anders möglich. Von daher war bei Ext:Commerce die Aufwandschätzung einfacher, weil man sich kein Gedanken machen muss, wo kann man was kaputt machen, überarbeitet man Kerneldaten; das sind Sachen, die man in TYPO3 oder Magento vermeiden muss. Auch wenn

das andere vielleicht nicht so ernst nehmen, ist es für mich in seiner klaren Qualitätsmerkmale, dass benutzerfreundliche und übersichtliche und auch die Klassen nicht über die einfachsten Methoden rufen (zum Beispiel in `pp/code/locator` komplette Klassen zu rufen), sondern dass man vernünftig Wege der Anpassung findet: am besten über Observer, Source-Modelle, Backend-Modelle oder ähnliches, oder zur Not noch über Reflection [Anm.: Service Locator], wenn es sich nicht vermeiden lässt. Solcher Mechanismen gibt es in TYPO3 vom Prinzip her auch, wobei sie mir in seinem Wissen nicht ganz so tief gehen wie bei Magento (aber das System kann ich auch nicht ganz so gut). Das hat sehr große Vorteile, hat aber auch die Nachteile, dass es komplizierter ist, Sachen damit zu entwickeln und seine Einstufige Schwierigkeit macht, also den richtigen Ansatzpunkt zu finden.

FS: Jetzt basieren alle drei auf PHP, auch wenn sie sehr unterschiedlich umgesetzt sind und PHP ist in der Sprache, die in meinem sehr viel Fröhlichkeit lässt, auch mal unsauber zu arbeiten. Macht sich das in Magento noch bemerkbar, obwohl es ja wie du sagst, in komplexen Systemen mit Anleihen aus der Software-Technik ist?

AvS: Das macht sich dann bemerkbar, wenn unerfahrene Entwickler damit arbeiten. Das habe ich in meinem ersten Projekt auch gemacht, weil ich einfach nicht herausfinden konnte, wie man die passenden Anforderungen umsetzen kann. Dafür fehlt einfach das Wissen und die Nachfragemöglichkeit. Wenn aber vernünftig Entwickler am Werk sind, halten sie die Fröhlichkeit dann doch in Grenzen. Man hat natürlich die Möglichkeit unsauber zu arbeiten, aber wer Magento einigermaßen vernünftig nutzt, der nutzt sie nicht, und das war in ext.commerce zum Beispiel ganz anders. Von daher bin ich schon sehr froh, dass Magento in für die PHP-Welt relativ starrs Korsett vorgibt, was dann in gewissermaßen Rahmen wirklich gute Entwicklung erzwingt. Man hat natürlich die Möglichkeit, sich darüber hinwegzusetzen, aber das merkt man dann auch sehr schnell.

FS: Man wird in Magento-Modulen also sehr typischerweise unsauber PHP-Konstruktionen haben, wie globale Variablen. Worum man aber nicht herumkommt sind Sprach-Eigenschaften von PHP wie zum Beispiel das schwache Typsystem. Beispielweise kann in Methoden unterschiedliche Typen zurückgeben und es lässt sich nicht festlegen, was für einen Typ in Variablen hat. Solcher Dinge kann das System natürlich nicht erzwingen...

AvS: Magento versucht das auch nicht. Manche Extension-Entwickler versuchen es und geben entsprechende Klassennamen für ihre Parameter vor [Anm.: „Type Hinting“ für Objekte als Funktions-/Methodenparameter ist die einzige Möglichkeit in PHP, Typen zu erzwingen] aber normalerweise wird das eher ignoriert oder es sogar ausgenutzt dass man diese Flexibilität hat. Das erfordert natürlich in gewisser Disziplin von den Entwicklern. Wenn in Methoden unterschiedliche Rückgabtypen haben kann, muss derjenige der die Rückgabverarbeiten auch alle diese Typen verarbeiten. Das muss man natürlich auch erstmal wissen und wenn man es nicht weiß oder nicht berücksichtigt, kann das auch zu weiteren Fehlern führen.

FS: Entsteht denn dadurch in der Praxis Konflikt?

AvS: Ja. Ich finde es angehen, so zu entwickeln, weil es manche Sachen einfach schneller macht aber man muss natürlich aufpassen.

FS: Die Lösung ist also „aufpassen und nachschauen“?

AvS: Ja, genau. Alles mit allen Konfigurationsmöglichkeiten durchtesten geht einfach nicht.

FS: Ist das etwas, was sich schon im Entwurf bemerkbar macht – wenn du da schon in den

Cod schau n musst – dass du darauf acht st, was s für v rschi d n Möglichk it n gibt. Od r sind das F inh it n, nach d n n du rst b im Programmi r n schau n musst?

AvS: Erst b im ntwick ln, das g ht zu s hr ins D tail als dass man s b i d r Konz ption b rücksichtig n könnt .

FS: Ein and r Schwäch von PHP, di häufig g nannt wird, ist dass s s hr vi l K rn-Funktion n gibt, di noch dazu s hr inkons qu nt in d r B n nnung und b i d n Param t rn sind.

AvS: Ja, das ist G wöhnungssach und da muss man infach imm r wi d r nachschlag n od r sich von d r Entwicklungsumg bung h lf n lass n.

FS: Das schafft auch zum B ispi l Mag nto nicht, zu abstrahi r n?

AvS: N in, man nutzt imm r wi d r K rnfunktion n, trotz Z nd Fram work, trotz Mag nto, darum kommt man nicht h rum. Das ist l tzt ndlich infach Erfahrungssach und ansonst n Fl ifarb it.

FS: Nachd m wir uns d n Hint rgrund PHP und PHP Fram works ang schaut hab n, inmal zu W b-Applikation n allg m in. Wir hab n uns bish r noch gar nicht damit b schäftigt, dass Mag nto ja in w bbasi rt s Syst m ist. Spi lt das in Roll od r würd st du sag n, b i in m Cli nt-S rv r-Syst m hätt man di s lb n Probl m bzw. würd sich d r Unt rschi d im Entwicklungsproz ss b m rkbar mach n?

AvS: Ich d nk das wär r lativ ähnlich. Wob i ich j tzt s it Jahr n k in D sktop-Syst m od r Cli nt-S rv r-Syst m ntwick lt hab , sond rn ausschli flich W b-Softwar . Ab r früh r hab ich da durchaus mit zu tun g habt und da s h ich j tzt k in groß n Unt rschi d .

FS: Was ist zum B ispi l damit, dass zur Ausführung imm r noch in Brows r b nötigt wird?

AvS: Ich v rsuch , so vi l wi möglich, üb r di Kommandoz il od r di Entwicklungsumg -bung zu t st n, zumind st w nn s kompl x r Funktion n sind, und di s rst spät r in das Us r Int rfac inzubau n.

FS: Erzähl doch mal, wi du das handhabst.

AvS: Zum B ispi l hab ich g rad vorg st rn in Export-Funktion für B st llung n impl -m nti rt. Ich hab damit ang fang n, di nötig n Mag nto-Mod ls, das nötig Modul, rst llt hab . Das hab ich ab r nicht dir kt ins Back nd int gri rt, sond rn hab mir in Sh ll-Skript g macht, als ang passt Kopi in r d r K rn-Dat i n und führ das auf d r Kommandoz il bzw. üb r di Entwicklungsumg bung mit d n notw ndig n Param t rn aus. So kann ich s auf Knopfdruck imm r wi d r ausführ n und b komm di Rückgab dir kt üb r di Kommandoz il od r schau mir di r sulti r nd XML-Dat i im T xt ditor an. Das g ht schn ll r, als würd man j d s Mal d n Brows r aufruf n und aktualisi r n.

FS: B i in r Export-Funktion macht das natürlich Sinn, würd st du s auch für and r Funktionalität mpf hl n?

AvS: Nur w nn s s hr kompl x Funktion n sind. Für all s was stark mit d m Us r Int rfac zu tun hat, dag g n nicht. Ab r zum B ispi l musst ich für di War nkorb-Darst llung von Produkt n di dahint r li g nd Coll ction anpass n und n usorti r n sowi zusätzlich F ld r hinzufüg n. Da konnt ich mir b ss r üb r in T xt-Ausgab ans h n, wi di Coll ction j tzt aussieht, ohn j d s mal d n War nkorb n u aufruf n zu müss n.

FS: Man könnte sagen dass du in Art manueller Unit Test machst, da du Funktionen isoliert vom User Interface testest.

AvS: Genau, im Prinzip mache ich in den automatisierten Tests, die ich mit dem eigenen Augen überprüfe.

FS: Wiesichtest du mit dem Debuggen aus?

AvS: Nutze ich mittlerweile relativ intensiv.

FS: Und funktioniert auch zufriedenstellend?

AvS: Das funktioniert sehr gut. Man braucht in den etwas stärkeren Bereichen, mein ist jetzt drei, vier Jahre alt, die braucht für in den Request dann schon mal in paar Sekunden. Aber in vielen Fällen ist es einfach unverzichtbar und man würde mit Debugging-Ausgaben oder Logging etc. die noch viel länger brauchen. Mit dem Debugger bekommt man auch in den besten Eindruck davon, was eigentlich genau passiert, zum Beispiel von welchen Klassen die aktuellen Code-Statements aufrufen wurden.

FS: Debugging ist also auch ein gutes Werkzeug um die Mechanik des Frameworks besser zu verstehen.

AvS: Genau. Und eschränkt die Magente-Entwickler, auch sehr erfahren, sehr zensiert nicht ein. Dabei ist die Hürde mittlerweile gar nicht so groß.

Man ist es einfach in der PHP-Welt nicht so gewohnt, mit dem Debugger zu arbeiten. Da haben die Leute die zum Beispiel mit Java oder C++ Erfahrung haben, den Einstieg deutlich einfacher. Ich denke sowieso, dass die einleichter den Einstieg in Magento finden als diejenigen, die aus der PHP-Welt kommen. PHP ist nur in anderer Programmiersprache mit einigen Besonderheiten aber die ganz Mechanismen und Techniken sind doch eher an Java u.ä. angelehnt. Mit der „oldstyle PHP Entwicklung“, die man seit 10 Jahren kennt, kommt man dagegen nicht so weit. Ich komme selbst aus dieser Richtung und habe auch die ersten zwei, drei Magente-Jahre nicht mit Debugger gearbeitet und wüßte keinen Einblick in die verschiedenen Design Patterns gehabt und so weiter und das hat den Einstieg schon schwieriger gemacht. Wenn man Erfahrung in der Java-Entwicklung hat, kommt man schon schneller in Magento rein.

FS: Da wundert man sich dann eher über die PHP-Eigenheiten als über das Framework...

AvS: Und über die Sachen, die viele PHP-Entwickler nicht machen, wie Debugging und automatisierte Tests. Dabei ist es eigentlich wirklich einfach, wenn man es einmal gemacht hat. Und wenn man die richtigen Software für in den lokalen Entwicklungsserver installiert und für die Entwicklungsumgebung, dann ist es auch praktisch in Konfigurationsarbeit.

FS: Ihr seid nun mit vier Leuten noch in der nächsten Team und du hast vorher auch schon lang allein oder in Kooperation mit anderen Freelancern gearbeitet. Was für Besonderheiten ergeben sich aus der Team-Größe? Gibt es Dinge, die man in größeren Teams besser handhaben könnte?

AvS: Ich würde fast sagen dass in kleineren Teams schlagkräftiger ist, weil man sich stärker fokussieren kann und weniger abstimmen muss. Bei größeren Teams müssen alle auf dem gleichen Stand sein und das ist dann eher Mehraufwand. Das heißt in kleineren Teams eher im Vorteil, vier Mann ist ein guter Größe für mein Geschmack. Natürlich ist höchstens, dass sie nicht ganz so viel schaffen können aber in Acht-Mann-Team schafft sich rielich nicht das Doppelte, eher das anderthalbfache. Man muss natürlich auch die richtigen Leute dabei haben

und wenn man in n dabei hat, der ntw d r auf Qualität nicht so viel Wert legt oder k in Mag nto-Erfahrung hat, dann ist s schon schwierig.

FS: Jetzt wird man s auch bei kl in n Teams h r s lt n schaffen, dass all s Exp rt n sind – grad auf in m G bit mit stil r L rnkurv im Sinn von schwierig m Einsti g.

AvS: Wir haben tatsächlich all mind st ns zw i inhalb Jahr Mag nto-Erfahrung und müssen uns über grundl g nd Sach n nicht mehr unterhalten. Wir wissen all , was die B st Practic s sind und dementsprechend funktionieren das auch. Das war auch in Motivation, die s s T am so aufzustellen.

FS: Du würdest also sagen, dass die s in wichtiger Faktor für den Erfolg kl in r T am s ist.

AvS: Für unser T am war das in wichtiger Faktor, man kann sich r auch mit unfahr n n L ut n im T am arbeits und s ist sich rlich in d r b st n Möglichkeit n, n u Entwickl r an Mag nto h ranzuführen, im T am mit rfahr n n Entwickl rn zusammenzuarbeiten. Aber das bedeutet natürlich auch mehr Aufwand, man kommt nicht so schnell voran und die Qualität wird v ntu ll nicht ganz so hoch s in außer man s tzt T chnik n wie Pair Programming, Reviews usw. in Das nutzen wir im Moment gar nicht, weil wir uns einfach gg ns itig so s hr v rtrauen und wissen, dass all die ntsprechend Erfahrung haben.

FS: Nun sind Reviews nicht nur dazu da, schwache T ammitglieder zu kontrollieren, sondern auch bei rfahr n n T am s in rprobt s Mittel.

AvS: Klar, das ist sicher noch etwas was wir machen könnten und die Qualität weiter steigern würdest aber für den Qualitätsanspruch den wir im Moment haben nicht unbedingt notwendig ist. In unserem Fall ist s daher auch h r in T am aus vier Einzelkämpfen, die all ihr Aufgaben haben und nachher die Sach n s i umg s tzt haben wird r zusammenschließen.

FS: Und die s s zusammenschließen funktionieren trotzdem gut?

AvS: Das funktionieren bisher ziemlich gut. Je nach Anforderung und Projekt ist s sicher sinnvoll, da ngr zusammenzuarbeiten aber da ist s häufig schon von Vorteil, im gleichen Raum zu sitzen und sich gg ns itig einfach mal fragen zu können.

FS: Wo siehst du im Entwicklungsprozess alles in allem am höchsten Verlässungsbedarf und -potential? Sowohl auf S it n d r G g b nh it n (Mag nto, verfügbar W rkz ug) als auch auf S it n d r Entwickl r?

AvS: Bei den Entwickl rn: mehr auf Qualität achten; die gg b n n W rkz ug auch nutzen (Debugging, Testen); Schulungsaufwand / Einarbeitungsaufwand in Kauf nehmen und nicht einfach drauf losarbeiten wie man s vill icht seit Jahren gewohnt ist; auch mal n u T chnik n ausprobieren, die in n dann auch weiterbringen; aktuell bleiben; informiert bleiben, darüber was s für n u Entwicklung n gibt.

Auf S it n d s Frameworks wäre häufig noch in b s s r Zusammenarbeit mit den Entwickl rn notwendig, also dass in off n r r Entwicklungsprozess stattfindet, dass s einfach r r ist für die Entwickl r, Cod b r itzustellen, zum Beispiel ige n Ev nts hinzuzufügen; dass das ganz etwas flexibler gehandhabt wird. Dazu müsste man natürlich mehr Personal bereitstellen, das für die Communitybetreuung da ist. Mehr Dokumentation auf jeden Fall, aber 2009 war das Jahr der Dokumentation [Anm.: Aussage von Mag nto Inc. 2008], 2010 wahrscheinlich auch... Da muss man einfach von Anfang an dran bleiben, wie s j tzt bei Mag nto 2 versucht wird. Da haben sie schon inig gut Ansatz , die si j tzt mit dem neuen Projekt

durchschnittlich. Es ist schon viel besser geworden, dadurch dass es Schulungen gibt, dass es Anwenderhandbücher gibt, dass es jetzt Zertifizierung gibt und dass jetzt mit Magento 2 versucht wird, viel transparenter zu gestalten und Contributions zu erleichtern. Aber da ist sicherlich noch Luft nach oben.

FS: Zu den Werkzeugen: du hast einige genannt, was die Entwicklung bereits vereinfacht, auch ein spezifisch Werkzeug – würdest du dir zusätzlich Werkzeug wünschen, die noch fehlen?

AvS: Schwere Sachen vorzustellen was es noch nicht gibt. Ich würde mir wünschen, dass es bei manchen Werkzeugen einfacher für nicht-Administratoren wäre, sie einzusetzen. Das geht vor allem in Richtung der Deployment-Werkzeuge oder wie in der Service-Mechanismen und Caching-Mechanismen: Varnish, Solr, Vagrant ... Werkzeuge, die das Deployment vereinfachen, das Aufsätzen der lokalen Entwicklungsumgebung, sind häufig auf Administratoren und Linux-Frameworks zugeschnitten und halten auch von solchen gemacht. Es gibt häufig Entwickler, die auch gleichzeitig gute Administratoren sind und genau solche Werkzeuge entwickeln oder einsetzen. Aber es gibt genauso gute Entwickler, die nicht gleichzeitig Sysadmins sind, zu denen ich mich auch zähle, und die dann Schwierigkeiten haben, solche Werkzeuge einzusetzen.

FS: Ist das Deployment auch ein generelles Problem? Zumindest aus Entwicklungssicht?

AvS: Dazu braucht man eigentlich einen Administrator, also aus Entwicklungssicht definitiv. Und es kann einfach nicht jeder sich die nötigen Fähigkeiten in beiden Bereichen aneignen.

FS: Liegt das auch wieder an der PHP-Umgebung? PHP hat zum Beispiel von sich aus schon einige Abhängigkeiten, benötigt externe Bibliotheken...

AvS: Dazu fehlt mir die Vergleichsmöglichkeit aber es ist definitiv ein Problem. Wenn man diverse Vorträge zum Thema Deployment oder „unser Werkzeug“ hört, die es zum Beispiel auf der Magento Imagine in Las Vegas gegeben hat: das waren 15, 20 Verschiedene Werkzeuge, die man einsetzen muss, die man als Entwickler dann auch beherrschen muss.

FS: Für Java gibt es zum Beispiel Maven, worin diverse Werkzeuge für den Software-Lebenszyklus zusammengeführt werden, da muss man eigentlich kein Experimentieren, weil in dem viel abgenommen wird.

AvS: Ich glaube, das ist Java auch systemunabhängiger, das funktioniert mit allen Systemen, ob es Windows, Macintosh oder Linux ist. Bei PHP ist das alles systemabhängiger. Ich glaube als Windowsnutzer hat man da noch mehr Schwierigkeiten als als Mac oder Linux Nutzer. In jedem Fall muss man sich auch mit Konfigurationsfiles und Kommandozeilen usw. herumschlagen und wirklich auch damit beschäftigt sein, sich teilweise Tag mit in dem Werkzeug beschäftigt sein bis man es beherrscht und vernünftig eingerichtet hat. Das ist etwas, was man definitiv verbessern könnte.

FS: Es gibt ja aktuell einige Ansätze in der PHP-Welt, zum Beispiel mit Composer für Packaging, oder ganz neu das PHP-Maven Projekt, was derzeit vorangetrieben wird. Würdest du sagen das ist der richtige Weg?

AvS: Das ist wahrscheinlich der richtige Weg, damit habe ich mich aber noch nicht aussprechen lassen.

FS: Das ist wahrscheinlich auch noch nicht alles für Magento einsetzbar ist.

AvS: Es gab auf dem letzten Hackathon in internet's Projekt, Magento mit Composer zu verhiert, so dass man sich in Modul über die Bibliothek kennt Modman [Anm.: Modul Manager] Files automatisch installiert lassen kann, mit ihrer Abhängigkeiten. Aber auch dafür habe ich das Gefühl, braucht man schon sehr viel Hintergrundwissen, um das vernünftig einsetzen zu können und muss mehrere Tools einsetzen. Das geht mir fast schon wider zu wissen um's mal benutzen ausprobieren. Da muss man sich dann wirklich zwei, drei, vier Tage hinsetzen bis man das eingrichtet hat, ist mein Eindruck.

FS: Abgesehen vom Deployment, ist die Werkzeugunterstützung für Magento OK?

AvS: Debugging, Entwicklungsumgebung, lokaler Server... das funktioniert alles mittlerweile sehr gut. Da gibt's eigentlich überall was, was out-of-the-box funktioniert und damit bin ich eigentlich ziemlich zufrieden.

FS: ... Test-Frameworks ausgenommen, mangels ausreichender Erfahrung damit ...

AvS: Ja. Es ist aber allgemein schon sehr viel besser geworden. Früher gab's sich auch schon Möglichkeiten, die ich nicht genutzt habe aber seitdem ich PHPStorm und Zend Server CE und den eingebaute Debugger benutze habe ich den ich doch deutlich Vorteil. Das sind relativ kleine Sachen, die man in einem halben Tag installiert hat, auch wenn man sich vorher nicht damit beschäftigt hat, und die machen das Entwicklungsbereich schon viel einfacher. Auch Kleinigkeiten wie das Magento-Plugin für PHPStorm machen viel noch einfacher.

FS: Eine Sache, auf die wir noch nicht im Detail eingegangen sind: du hast am Anfang das Problem der Flexibilität angesprochen, wenn zum Beispiel eine Anforderung neuerinkommen. Gibt's irgendwelche konkreten Maßnahmen um das aufzufangen oder ist das in sich aufgelöst? Das Problem, das zumindest in dem Prozess an der ihr derzeit gebunden seid, nicht besser zu machen ist?

AvS: Ich glaube mit dem Wasserfallmodell ist das nicht lösbar. Da wird's immer Anforderung gebogen, die erst im Nachhinein auftauchen und dann halt in den CRVfordern, dafür muss man halt wirklich in anderen Modellen hängen.

FS: Hast du in vorigen Projekten Erfahrung mit agilen Modellen gemacht?

AvS: Also nicht ernsthaft mit Scrum o.ä. aber halt einfach die Bezahlung nach Aufwand, und dass man kein vorgeschriebenes Anforderungsdokument hat. Das habe ich schon mehrfach mitgemacht und das ist für mich das deutlich angenehme Arbeiten und auch das was nachher der Kunde anforderung entspricht und häufig auch günstiger für den Kunden ist, wenn man den Kunden ein Puffer einrechnen muss. Den Puffer braucht man als unerfahrener Entwickler zwar auch häufig oder überschränkt ihn sogar aber als erfahrener Entwickler bleibt er doch meist ungenutzt.

FS: Gibt's sonst noch etwas, was du loswerden willst, was ich vielleicht nicht abgedeckt habe?

AvS: Vielleicht noch etwas zu Versionskontrollen. Für mich ist's immer noch faszinierend, dass es Entwickler gibt, die überhaupt kein Versionskontrollen einsetzen, denn das ist für mich das absolute Grundkriterium dass man überhaupt so etwas wie Qualität haben kann; dass man einigermaßen sicher sein kann, dass das was man heute programmiert auch morgen noch da ist. Es gibt erstaunlich viele, die das immer noch nicht einsetzen. Ich habe jahrelang mit SVN gearbeitet und war sehr zufrieden, jetzt arbeite ich mit Git, bin ein bisschen zufriedener

aber auch nicht hundertsprozentig überzogen, weil es widerin großes Wissen erfordert um sicher richtig zu beherrschen.

FS: Was fehlt da?

AvS: Da fehlt wider der Fokus auf die nicht-Experimente, die sich nicht in Wochen nur damit beschäftigt haben, alle Details, auch der Implementierung, von Git kennenzulernen, sondern auch der Leute die sich einfach nur für ihr tägliches Versionsverwaltung insitzen wollen und nicht viel mehr machen wollen als commit, update, und vielleicht mal in neuen Branch einrichten und mergen.

FS: War das mit SVN besser?

AvS: Ja, das war intuitiver zu bedienen. Es war zwar langsamer und schwefälliger und das Branching funktioniert nicht vernünftig, aber dafür konnte man deutlich weniger kaputtmachen, hatte ich das Gefühl. Dafür ist es mit Git jetzt tatsächlich öfter passiert, dass Code einfach weg war, vor allem wenn man mit Leute arbeitet, die sich noch nicht so lang insitzen. Da müssen dann alle das System beherrschen und wissen was sie tun und was sie über sich und was nicht. Deswegen bin ich da auch nicht hundertprozentig glücklich damit.

FS: Wie ist es mit der Integration des Magento-Kerns ins Versionsmanagement? Es ist ja nun mal so dass der originale Code oft in Verzweigungen steht, die schon zu Magento gehören, beispielsweise Templates in `pp/design`. Wie handhabt man das?

AvS: Da gibt es mehrere Vorgehensweisen. Entweder man nimmt den Magento Core direkt hinein, also alle Dateien bis auf ein paar in die .htaccess oder die Dateien mit den Datenbank-Zugangsdaten, sind im Repository. Die andere Möglichkeit ist, den Kern mit im Entwicklungszweig zu haben, aber nicht im Repository. Das funktioniert mit der Entwicklungsumgebung auch ganz gut, dass wirklich nur die Dateien, die von niemandem angelegt werden, im Repository landen und alles andere ignoriert wird. Also nicht über die .gitignore Funktion, da man häufig Probleme hat, dass man ganz Verzweigungen ignoriert in die man irgendwann doch mal in die schreiben muss. Es wird letztlich einfach nicht dem Repository hinzugefügt. Und es gibt noch die dritte Möglichkeit, dass der Code-Bestandilerst an der Stelle zusammengefügt werden wo der lokale Workspace auch drauf zugegriffen und dass der Core-Repository unabhängig ist von dem Modul-Repository oder dem Projekt-Repository und das über irgendeinen Mechanismus zusammengeführt wird, sei es über die Entwicklungsumgebung und der Deployment-Funktion oder sei es über ein Tool wie Modman. Und das geht auch ganz gut, da habe ich wenig Probleme festgestellt.

FS: Es existieren also aussergewöhnliche Workarounds.

AvS: Genau, das macht jeder in bisschen anders und ich habe da schon mehrere funktionierende Methoden kennengelernt.

FS: Damit sind wir durch. Dank für das Interview!

A.1.3 Paraphrasiertes Interview mit Überschriften

Projektmanagement

Thema: Prozessmodelle

Frage: Was sind die Erfahrungen mit dem Wasserfall-Modell?

Das würd ich nicht mehr ins tzen, wenn es vermeidbar ist. Es ist zwar theoretisch planbar, funktioniert in der Praxis aber nur bei bestimmten Gründen:

- Man wird unflexibel: Auf Anforderungsänderungen kann nicht flexibel reagiert werden (Change Request (CR) stoppt den gesamten Prozess von vorn in Gang), man muss mit festem Budget arbeiten.
- Zeitplanverzögerungen durch jeden Change Request oder bei unvorhergesehenen Ereignissen (aktuelle Basisplan: warten auf Daten für Import-Schnittstelle)

Wir würden uns in flexibleren Modellen wünschen, das zumindest bei Bezahlung nach Aufwand und regelmäßigem Überprüfung der zu erledigenden Arbeiten enthalten sollte.

Frage: Gibt es konkrete Maßnahmen, um die fehlende Flexibilität aufzufangen oder ist das ein ungelöstes Problem, das in diesem Prozessmodell nicht besser zu machen ist?

Ich glaube, das ist mit dem Wasserfall-Modell nicht lösbar. Es wird immer Anforderungen geben, die erst im Nachhinein auftauchen und dann halt in den Change Requests, dafür müsste man also wirklich in anderen Modellen helfen.

Frage: Hast du in vorigen Projekten Erfahrung mit agilen Modellen gemacht?

Nicht ernsthaft mit Scrum oder ähnlichem aber zumindest die Bezahlung nach Aufwand und dass man kein vorgeschriebenes Anforderungsdokument hat. Das habe ich schon mehrfach mitgemacht und ist für mich angenehm, entspricht nachher den Kundenanforderungen und ist auch häufig günstiger, weil man kein Puffer einrechnen muss, der bei der Entwicklung doch meist ungenutzt bleibt.

Frage: Abgesehen von den Problemen, die das Wasserfall-Modell von Haus aus mit sich bringt: Gibt es dabei Probleme, die spezifisch für Web-Applikationen oder Magento sind?

In unserem Angebot steht zum Beispiel, dass das Layout nicht zu sehr vom Standard-Layout abweichen darf. Magento liefert ein Standard-Layout als Grundgerüst, darin sind ca. 20-40 verschiedene Siten typen. Die alle individuell umzusetzen würd jedes Budget überschreiten. Wir müssten allerdings bereits in Budget angeben, ohne das fertige Layout zu kennen und deshalb bei Bestellung und der gelieferten Layouts jedes Mal auf Konformität zu unserem Angebot prüfen.

Thema: Zusammenarbeit zwischen Agenturen

(Fortsetzung)

Wir haben noch die Besonderheit, dass wir als Unterauftragnehmer für in andere Agenturen arbeiten und keinen direkten Kontakt zum Kunden haben. Wird in andere Agentur ist mit der Erstellung des Layouts beauftragt, aufgrund dessen wir dann die Templates umsetzen.

Wir durchlaufen da aus den oben genannten Gründen derzeit im hr r Schließen und erst wenn wir alles abgecheckt haben, geht das Layout zum Kunden und er kann es abnehmen. Das ist in komplizierten Ablauf, der Ressourcen frisst. Die Überprüfung von fünf Layouts hat gerade wieder die Leute anderthalb Stunden beschäftigt.

Frage: Ist diese Zusammenarbeit von verschiedenen Agenturen eine typische Konstellation?

Das habe ich schon mehrfach erlebt und ist in einigen Projekten der Fall.

Frage: Und diese Abhängigkeit von einer anderen Agentur, die parallel das Layout entwickelt so dass mitten im Ablauf konkrete Anforderungen entstehen, macht die das ganze Modell kaputt?

Ja, es wird starr und man muss Schließen der um die Layoute dazu zu bekommen, die Site so zu machen dass sie uns kein Mehraufwände verursachen. Die Kontrollen aber verursachen selbst Mehraufwände, die im Ursprungsprojektpreis nicht mit kalkuliert waren.

Frage: Ist das generell ein Problem oder spielen da aktuell noch andere Faktoren hinein?

Das ist besonders durch den Festpreis in Problem. Bei der Bezahlung nach Aufwand habe ich bessere Erfahrung gemacht, da wird zwar bei steigenden Anforderungen der geschätzt Aufwand deutlich überschritten aber das Zusammenarbeiten ist angenehmer und das Ergebnis ist auch sehr gut. Ich arbeite grundsätzlich lieber nach Aufwand, es erfordert allerdings dass der Kunde flexibel ist und sich auch auf das Modell einlässt.

Thema: Teamgröße

Frage: Ihr seid mit vier Leuten noch ein recht kleines Team und du hast vorher auch schon lange alleine oder in Kooperation mit anderen Freelancern gearbeitet. Was für Besonderheiten ergeben sich durch die Team-Größe? Gibt es Dinge, die man in größeren Teams besser handhaben könnte?

Im Gegenteil, in kleineren Teams ist schlagkräftiger, man kann sich stärker fokussieren und muss sich weniger abstimmen. Bei größeren Teams bedeutet es mehr Aufwand, sich zu stellen dass alle auf dem gleichen Stand sind. Vier Mann ist für mich in der Geschmack in guter Größe, sie schaffenவில் nicht ganz so viel aber in Acht-Mann-Team schafft sich wirklich nicht das Doppelte, eher das anderthalbfache.

Voraussetzung ist, dass man die richtigen Leute dabei hat; wenn jemand dabei ist, der nicht auf Qualität nicht so viel Wert legt oder kein Magister-Erfahrung hat, dann wird es schon schwierig. Wir haben alle mindestens zwei innerhalb Jahr Magister-Erfahrung, kennen die Best Practices und dementsprechend funktioniert das auch. Für unser Team war das in Motivation und ein wichtiger Erfolgsfaktor.

Man kann auch mit unerfahrenen Leuten im Team arbeiten und es ist sicher in der besten Möglichkeit, neue Entwicklungen anzuführen, mit unerfahrenen Entwicklern im Team zu arbeiten. Es bedeutet aber auch mehr Aufwand, man kommt langsam voran und die Qualität ist vermutlich nicht ganz so hoch, außer man setzt Techniken wie Pair Programming, Reviews usw. ein. Darauf verzichtet man nicht, weil wir uns gegenseitig so sehr vertrauen und die entsprechenden Erfahrung haben. Es würde die Qualität sicher noch weiter steigern, das ist aber für unser derzeitigen Anspruch nicht unbedingt notwendig. Wir sind also quasi

in T am aus vi r Einz lkämpf rn, di ihr j w ilig n Aufgab n hab n und ihr Erg bniss zusamm nschm iß n.

Frage: Und dieses Zusammenschmeißen funktioniert gut?

Ja, j nach Anford rung und Proj kt wär s zwar sinnvoll, ng r zusamm nzuarb it n ab r s ist häufig schon von Vort il, im gl ich n Raum zu sitz n und sich g g ns itig infach mal frag n zu könn n.

Konzeption/Spezifikation

Thema: Anforderung; Spezifikation
--

Frage: Selbst als erfahrener Entwickler weißt du oft noch nicht genau, was es z.B. für Events gibt, die man nutzen könnte, wenn du die selbe oder eine ähnliche Anforderung nicht schon einmal umgesetzt hast. Ist das richtig und falls ja, stellt das beim Entwurf ein Problem dar?

Es ist natürlich in Probl m, w nn man b im Entwurf noch nicht g nau w iß, wi man twas umsz n kann. Es b st ht dann imm r rst in R ch reh -Schritt.

Frage: Wie läuft denn bei euch so eine Modulentwicklung ab? In der Theorie würde man beim Wasserfall-Modell den kompletten Entwurf fertigstellen bevor man anfängt zu codieren.

Man b kommt zunächst in Anford rungsdokum nt. W nn ich auf d r Basis in n Pr is abg b n soll, mach ich mir schon in d m Mom nt G dank n zur Ums tzung, schr ib dazu di inz ln n T ilanford rung n h runt r und b n nn zu d n inz ln n Punkt n in n Pr is.

W nn ich nicht schon g nau w iß wi ich vorg h , w il ich in ähnlich Anford rung b r its in in m and r n Proj kt umg s tzt hab , muss ich dazu schon in rst R ch reh -Rund inlg n: schau n wo s pass nd Ev nts gibt, w lch Klass n ang passt w rd n müss n usw. – dab i kommt s auch vor, dass ich b r its Sach n im Cod ausprobi r . Ich muss hi r also vorgr if n und hab zu früh schon vi l mit D tails zu tun. Danach hab ich sozusag n schon in Ums tzungssp zifikation f rtig, s ist ab r ig ntlich zu vi l Arb it zu di s m Z itpunkt.

Frage: Gibt es noch andere Probleme, die bei der Spezifikation auftreten?

Es ist in allg m in s Softwar -Probl m, dass di Anford rung n häufig nicht sp zifisch g nug sind, nicht durchdacht g nug, dass vi l D tailfrag n noch g klärt w rd n müss n, di t ilw is in n größ r n Einfluss auf d n Aufwand hab n.

B i Anw ndungsfram works wi Mag nto kommt t ilw is hinzu, dass Kund n zu w nig K nntnis von d r Grundanw ndung hab n und davon, w lch F atur s schon nthalt n sind. Es gibt Kund n, di Schritt für Schritt b schr ib n, was in Shop könn n muss, da kommt s dann vor, dass ganz infach , unwichtig Sach n vom Mag nto-Standard abw ich n und m hr r Tag Aufwand v rursach n würd n. Es ist häufig nicht ganz infach, d m Kund n so twas auszurd n.

Aus d m Grund, dass k in Anford rung n b r chn t und ggf. n u ntwick lt w rd n sollt n, di b r its im Standard nthalt n sind, ist s auch für di Entwickl r wichtig, das ganz Fram work und s in F atur s zu k nn n, und imm r auf d m lauf nd n zu bl ib n üb r Updat s ab r auch üb r Modul und Entwicklungshilf n aus d r sog nannt n Community.

Als Beispiel für Entwicklungshilfen das relativ neue PHPStorm Plugin Magicnto, das die Entwicklung deutlich vereinfacht.

Thema: Aufwandschätzung

Frage: Wie gut funktioniert die Aufwandschätzung bei Magento-Projekten generell?

In den ersten ein, zwei Jahren hatte ich oft das Problem, dass meine Aufwandschätzung deutlich zu gering war, da ich manche Magento-Probleme nicht einschätzen konnte oder nicht im Vorhinein gesehen habe. Aber seit ich zwei, drei Jahre Magento-Erfahrung habe und das System auch entspricht von innen kennen, habe ich meine Schätzungen immer weitgehend eingehalten, sofern keine neuen Anforderungen kommen sind.

Frage: Man muss das System dazu also auch „von innen kennen“ und nicht bloß damit gearbeitet haben?

Es hilft natürlich auch sehr wenn man ähnliche Anforderungen bereits umgesetzt hat. Viel, die noch nicht so lang mit Magento gearbeitet haben, unterschätzen das System und die anfallenden Aufwände. Das System ist extrem flexibel, die Flexibilität zu bewahren stellt aber auch erhöhte Anforderungen an den Entwickler.

Architektur/Entwurf

Thema: Magento-Architektur

Frage: Das genannte Magicnto-Plugin adressiert Probleme, die aus der Magento-Architektur herrühren, was sind denn das für Probleme?

Zunächst die Factory-Methode: Wenn man sich mit dem System nicht auskennt, ist es schwierig, herauszufinden, welche Klasse z.B. `core/log/product` ist und entspricht wo die Methode den desired Objekten zu finden sind. Da gibt es Möglichkeiten, die der Entwickler erstmal kennen muss und Umweg, die er gehen muss, und mit Magicnto ist das deutlich besser geworden, das Plugin erkennt die richtige Klasse und ermöglicht es, dort hin einzuspringen.

Ähnlich ist es mit den XML-Datien, für die kin Schema-Definition existiert, weil sie sehr für die aufbaut werden können. Das Plugin unterstützt nun trotzdem Shortcuts und Autocompletion bei der Erstellung der XML-Konfiguration.

Dies sind allerdings Probleme, die vor allem Anfänger vor größeren Hürden stellen, mit der Erfahrung hingegen sind sie gering.

Das Templating-System von Magento ist auch nicht ganz einfach, weil es durch die Layout-XMLs gesteuert wird. Um das richtig und inigermassen updatieren zu machen, darf man nicht das ganze Template überschreiben oder gar das Kern-Template anpassen, sondern nur die nötigen Template-Datien ins eigene Theme packen. Genauso sollte kein Layout-XML-Datien überschreiben werden sondern nur die Änderungen in eigenen Datien packen werden (z.B. die `local.xml`). Wenn man das sauber hinbekommt, habe ich selber erst sehr spät herausgefunden.

Implementierung

Thema: Standards (Code-Qualität)

Frage: Magento ermöglicht ziemlich viel Flexibilität. Mancher würde sagen zu viel, da es verschiedene Möglichkeiten gibt, ein Ziel zu erreichen und die für Anfänger vielleicht am einfachsten aussehende ist, einfach Klassen zu überschreiben. Ist das der Fall?

Ich verfolge ichs immer gerne mit Typo3, dort gab es viele mehr Möglichkeiten und das fand ich eher noch schlüsslicher. Bei Magento gibt es häufig in Ordnung, in Anforderung vernünftig umzusetzen, also inigmaßen standardkonform und upgradefähig. Ein Vorteil daran ist auch, dass man für modulare nicht überblicken kann und sehr schnell sieht, was sie machen, indem man sich nur die Konfigurationsdatei ansieht.

Frage: Du sagst „standardkonform“, sind denn die gegebenen Standards ausreichend, dokumentiert, brauchbar?

Dokumentiert sind die nicht vernünftig. Am besten kann man sich die Standards aus dem Kernmodul holen. Man hat sie dann richtig, wenn die eigenen Modul ähnlich aussieht wie im Kernmodul. Das sind ganz gute Beispiele, nach denen man sich richten sollte und die auch allgemein als Best Practice angesehen werden.

Es ist dazu in der Praxis, sehr viel Copy & Paste einzusetzen, also nachzuschauen, welche Module in ähnlicher Funktionalität hat (z.B. nur Tab im Benutzerkonto: Wunschlistenmodul) und zu schauen, was genau das macht. Die entsprechenden Titel kopiert man sich, passt die Identifizierung und setzt sie in die eigenen Code dazu. Man muss nur wissen, wo sie jeweils steht.

Thema: PHP

Frage: PHP ist eine Sprache, die einem sehr viele Freiheiten lässt, auch mal unsauber zu arbeiten. Macht sich das in Magento bemerkbar, obwohl es ja wie du sagst, ein komplexes System mit Anleihen aus der Softwaretechnik ist?

Das macht sich dann bemerkbar, wenn unerfahrene Entwickler damit arbeiten. Wenn vernünftig Entwickler am Werk sind, halten sie die Freiheit in Grenzen. Man hat natürlich die Möglichkeit, unsauber zu arbeiten, aber man nutzt sie nicht, das war in der Community zum Beispiel ganz anders. Von daher bin ich schon froh, dass Magento in für die PHP-Welt relativ starrs Korsett vorgibt, was in gewissermaßen Rahmen gut Entwicklung erzwingt. Man hat zwar die Möglichkeit, sich darüber hinwegzusetzen, aber das macht sich auch schnell bemerkbar.

Frage: Dazu kommen Sprach-Eigenheiten von PHP wie das schwache Typsystem, um die man nicht herum kommt.

Magento versucht auch nicht, Variablen- und Rückgabtypen zu erzwingen. Manche Extension-Entwickler versuchen sie und geben entsprechende Klassennamen für ihr Parameter vor [Anm.: „Typ Hinting“ für Objekte als Funktions-/Methodenparameter ist die einzige Möglichkeit in PHP, Typen zu erzwingen] aber normalerweise wird das eher ignoriert oder es sogar ausgenutzt, dass man dies Flexibilität hat. Das erfordert natürlich in gewisser Disziplin von den Entwicklern. Wenn in Methodenverschiedene Rückgabtypen haben kann, muss derjenige, der die Rückgabverarbeitet auch alle diese Typen verarbeiten. Das muss man natürlich auch erst mal wissen und wenn man sie nicht weiß oder nicht berücksichtigt, kann

das auch zu w it r n F hl rn führ n. Trotz d m find ich s ang n hm, so zu ntwick ln, w il s manch Sach n infach schn ll r macht. Man muss all rdings aufpass n.

Frage: Die Lösung ist also „aufpassen und nachschauen“?

Ja, g nau. All s mit all n Konfigurationsmöglichk it n durcht st n g ht infach nicht.

Frage: Macht sich dieses Konfliktpotential schon im Entwurf bemerkbar?

N in, das g ht zu s hr ins D tail.

Frage: Eine andere Schwäche von PHP, die häufig genannt wird, ist dass es sehr viele Kern-Funktionen gibt, die noch dazu sehr inkonsequent in der Benennung und bei den Parametern sind.

Ja, das ist G wöhnungssach und da muss man infach imm r wi d r nachschlag n od r sich von d r Entwicklungsumg bung h lf n lass n.

Frage: Das schafft auch z.B. Magento nicht, zu abstrahieren?

N in, man nutzt imm r wi d r K rnfunktion n, trotz Z nd Fram work, trotz Mag nto, darum kommt man nicht h rum. Das ist l tzt ndlich infach Erfahrungssach und ansonst n Fl ißarb it.

Thema: Debugger

Frage: Funktioniert das Debugging zufriedenstellend?

Das funktioni rt s hr gut. Man b nötigt für d n D bugg r in n twas stärk r n R chn r ab r mit D bugging-Ausgab n od r Logging würd man noch vi l läng r brauch n. Und in vi l n Fäll n ist s infach unv rzichtbar, mit d m D bugg r b kommt man in n b ss r n Eindruck davon, was g nau g rad passi rt, z.B. von w lch n Klass n di aktu ll Cod -St ll aufg ruf n wurd : Ein gut s W rkz ug, auch um di M chanik d s Fram works b ss r zu v rst h n.

Ab r rschr ck nd vi l Mag nto-Entwickl r, auch s hr rfahr n , s tz n s nicht in, obwohl di Hürd mittl rw il gar nicht so groß ist. Mit d r richtig n Softwar für d n lokal n Entwicklungss rv r und di Entwicklungsumg bung ist s auch praktisch k in Konfigurationsarb it. Das ist in Probl m aus d r PHP-W lt, man ist s dort nicht g wohnt, mit in m D bugg r zu arb it n [Anm rkung: Erst mit d r V röff ntlichung von Xd bug im Jahr 2002 – Qu ll : <https://github.com/xdbug/xdbug/blob/f22adf716b1aa069d07674bb914234992a212c/NEWS> – wurd in int raktiv r D bugg r für PHP ing führt]

Thema: Versionskontrolle

Frage: Gibt es sonst noch etwas wichtiges zur Implementation?

Es gibt rstaunlich vi l Entwickl r, di gar k in V rsionskontroll ins tz n, für mich ist si in Grundkrit rium für Qualität. Ich hab jahr lang mit SVN g arb it t, j tzt mit Git, b id hab n ihr Vor- und Nacht il . SVN war langsam r und schw rfällig r und das Branching hat nicht richtig funktioni rt, dafür konnt man w nig r kaputtmach n, s war zugänglich r für nicht-Exp rt n, di nicht vi l m hr mach n woll n als commit, updat und vi ll icht mal branch und m rg .

Frage: Wie funktioniert die Integration des Magento-Kerns ins Versionsmanagement? Eigener Code steht ja häufig in Verzeichnissen innerhalb des Kernsystems.

Da gibt es mehr verschiedene, die zufriedenstellend funktionieren. Man kann den Core komplett ins Repository mit hinein nehmen, ihn im Entwicklungsvorgang halten und ihn ins Repository zu integrieren, oder die Code-Basis dort zusammenfügen, wo der lokale Workflow darauf zugreift und in Core-Repository getrennt von Modul- und Projekt-Repository pflegen. Tools wie Modman oder Deploy-Funktion der Entwicklungsumgebung unterstützen bei der Zusammenführung. Da habe ich bisher wenig Probleme festgestellt.

Test

Thema: Funktionale Tests: Manuell, Selenium

Frage: Gerade bei Updates und bei der Änderung von übernommenen Shops will man sicherstellen, dass bereits vorhandene Funktionalität bestehen bleibt. Wie lassen sich Regressionstests dazu umsetzen und was gibt es dabei für Probleme?

Bisher habe ich hauptsächlich manuell getestet, das klappt nur bedingt. Man kann als Einzelperson oder Test-Team so nicht alle Funktionen kontinuierlich durchtesten. Bei einem Update von 1.3 auf 1.7 haben wir z.B. erst einen Monat später entdeckt, dass die Vergleichsfunktion nicht mehr funktioniert, weil in der Template-Datei. Da wünscht man sich automatisierte Tests, doch die gibt es nicht und sie zu erstellen ist ein großer Aufwand, den kaum ein Kundenzahlmeister möchte.

Es gibt seit einiger Zeit das Magento Test Automation Framework, das Standardfunktionen mit Selenium testet. Das funktioniert natürlich nur, wenn kein relevanter Änderung an der Seite vorgenommen wurden oder das Testframework von Anfang an integriert wird und Stück für Stück an die jeweiligen Anpassungen angepasst wird. Mir fehlt bisher immer die Möglichkeit oder das Budget dazu.

Ein Nachteil ist, dass auch so ein Testlauf mehrere Stunden dauert und der kontinuierlich zu testen der Bereich gegebenenfalls eingegraben werden müsste. Und solche Selenium-Tests kann man auch eigentlich nur am Live-System sinnvoll anwenden, sodass ich denke, man hat ein Testsystem, bei dem sichergestellt ist, dass es auf dem neuesten Stand ist.

Frage: Kommt es deiner Erfahrung nach denn häufig vor, dass Modul X in Modul Y oder dem Magento-Kern unerwartete Seiteneffekte hat, die man wahrscheinlich erst mit einem automatisierten Test unmittelbar gefunden hätte?

Ja, besonders bei komplexen Änderungen, z.B. beim Kundenmodell, wo man vermutlich nicht immer alle Spezialfälle im Überblick hat. Wenn man dann irgendwann Magento-Standardfunktionen nutzt, die bei der Entwicklung der Zusatzfunktion nicht berücksichtigt wurden (z.B. die Flat Product Tables), dann können Bugs an Stellen auftreten, an denen man sie nicht vermutet hätte.

Thema: Unit Tests: Manuell, PHPUnit Extension

Frage: Jetzt hast du, wie ich weiß, mittlerweile auch Erfahrung mit Unit Tests. Wie sieht es damit aus, gibt es da bezüglich Magento Probleme?

Grundsätzlich denke ich, dass es sinnvoll ist, Unit Tests anzunehmen, zumindest bis zu einer gewissen Ebene, wo man dann auf funktionale Tests umschwenkt. Ich bin allerdings daran gescheitert, dass ich mich mit den Tests noch nicht so gut auskenne und niemanden hatte,

den ich da ansprechen konnte, und schwierig Dokumentation zu den beiden auf PHPUnit basierender Magento Unit Test Frameworks gibt, im meinem Fall EcomDev PHPUnit von Ivan Churnyi. Ich hatte das Problem, dass ich sehr erfahren mit Magento bin aber sehr wenig erfahren mit Unit Tests. Wenn in Testfall schlug, lag das dann meistens am Test, nicht am Produktionscode. Das hat mich Unsicherheit hineingebracht, und ich bin zum Schluss gekommen, dass es für mich nichts bringt – nicht ohne Schulung für einen fundierten Einstieg – abgesehen von einfachen Basis-Tests (existiert in Layout-Datei, ist in Observer korrekt definiert [Anmerkung: das sind wichtige Assertions im o.g. Framework]).

Das Problem ist vor allem das komplizierte Fixture-System: wie bekomme ich richtige Testdaten, zum Beispiel in korrekter konfigurierter Produkt, wie ich es auch im Backend anlegen oder importieren würde. Das ist wieder in Problem der Dokumentation, es gibt nicht genügend Beispiele, so dass ich bei Spezialfällen nicht mehr weiter wusste. Für einfache Module kann ich es mir herleiten, bei komplexeren Abhängigkeiten ist die Dokumentation zu dünn. Wie die Daten in der Datenbank aussahen würde, weiß ich, aber nicht, wie ich die entsprechend Fixture schreiben muss.

Frage: Gibt es Alternativen zu Unit Tests?

Ich versuche, so viel wie möglich, über die Kommandozeile oder die Entwicklungsumgebung zu testen, zumindest wenn sie komplexe Funktionen sind, und dies erst später in das User Interface einzubauen.

Zum Beispiel habe ich eine Export-Funktion für Bestellungen implementiert und das Modul zunächst nicht ins Backend integriert sondern über ein Shell-Skript mit den notwendigen Parametern aufrufen. Die Rückgabewerte bekomme ich direkt auf der Kommandozeile bzw. ohne Umweg über den Browser als XML-Datei, die ich mir im Texteditor anschauen kann.

Ich empfehle die selbigen Vorgehen für komplexe Funktionen, die nicht stark mit dem User Interface zusammenhängen. Für einen Warenkorb-Darstellung muss ich zum Beispiel die Produkt-Collection aufrufen, das konnte ich mir besser über die Text-Ausgabe anschauen als jedes Mal den Warenkorb aufzurufen.

Frage: Man könnte also sagen, dass du eine Art manuellen Unit Test machst, da du Funktionen isoliert vom User Interface testest?

Genau, im Prinzip in den automatisierten Tests, den ich mit dem eigenen Auge überprüfe.

Integration

Thema: Konfiguration

Frage: Du sagtest, das Testsystem für funktionale Tests muss genau auf dem selben Stand sein. Externe Einflüsse wie die Systemkonfiguration haben also auch noch Auswirkungen?

Genau. In unserer aktuellen Projektarbeit mehrerer Entwickler aus verschiedenen Firmen. Da sind wir dazu übergegangen, sämtliche Konfigurationsänderungen nur noch über ein Update-Skript im Modul vorzunehmen, nicht über die Anwendung. Das erfordert erstmal mehr Aufwand, sorgt aber als qualitätssichernde Maßnahme dafür, dass sämtliche Instanzen auf dem selben Stand sind: Kategorien, CMS-Siten etc. – alles über PHP und SQL Code.

Frage: Das betrifft jetzt die Magento-Konfiguration, Nun ist es gerade bei PHP ja so, dass auch noch viel von der Systemkonfiguration abhängig ist, wie der globalen php.ini. Ist das ein Problem oder wird das vom Framework ausreichend abstrahiert?

Viel abstrahieren das, indem sie auf allen Entwicklungsinstanzen die gleiche Umgebung inszenieren. Ich mache das nicht, habe damit aber auch wenig schlechte Erfahrung gemacht. Mir ist schon passiert, dass es Groß-/Klein-schreibungsprobleme gab, weil ich Windows inszenierte und das System Groß- und Klein-schreibung ignorierte, Linux auf dem Server dagegen nicht. Aber das ist relativ selten. Es kann auch passieren, dass es Synchronisationsprobleme z.B. durch unzureichende Speicherlimit gibt, mein Monitoring nach ist in synchronisierter Umgebung aber nur bei großen Projekten sinnvoll, wenn man nicht bereits in vorbereiteter Umgebung hat, die für alle Projekte verwendet wird.

Thema: Deployment-Werkzeuge

Frage: Was würdest du dir bezüglich der Werkzeugunterstützung wünschen?

Ich würde mir wünschen, dass es bei manchen Werkzeugen einfacher für nicht-Administratoren wäre, sie einzusetzen. Das betrifft vor allem Deployment-Werkzeuge oder weitere Server-Mechanismen und Caching-Mechanismen. Diese sind häufig auf Administratoren zugeschnitten und auch von solchen gemacht. Entwickler wie ich, die nicht gleichzeitig Systemadministratoren sind, haben dann Schwierigkeiten, solche Werkzeuge einzusetzen.

Frage: Ist das Deployment auch ein generelles Problem? Zumindest aus Entwicklersicht?

Aus Entwicklersicht definitiv, dazu braucht man eigentlich einen Administrator. Und es kann sich nicht jeder die nötigen Fähigkeiten in diesem Bereich aneignen.

Frage: Liegt das auch wieder an der PHP-Umgebung? PHP hat z.B. von sich aus schon einige Abhängigkeiten, benötigt externe Bibliotheken...

Dazu fehlt mir die Vergleichsmöglichkeit aber es ist definitiv ein Problem. Wenn man diverse Vorträge zum Thema Deployment oder „unser Werkzeug“ hört, die z.B. auf der Magento Imagin in Las Vegas gebunden hat: das waren 15, 20 verschiedene Werkzeuge, die man inszenieren muss, die man als Entwickler dann auch beherrschen muss.

Frage: Zum Vergleich: Für Java gibt es z.B. Maven, worin diverse Werkzeuge für den Software-Lebenszyklus zusammengeführt werden, da muss man eigentlich kein Experte sein, weil einem vieles abgenommen wird.

Ich glaube, das ist Java auch systemunabhängiger, das funktioniert mit allen Systemen, ob es Windows, Macintosh oder Linux ist. Bei PHP ist das alles systemabhängiger, gerade als Windowsnutzer hat man da Schwierigkeiten. In jedem Fall muss man sich mit Konfigurationsfiles und Kommandozeilensyntax usw. herumschlagen und sich teilweise Tag für Tag intensiv mit dem Werkzeug beschäftigen bis man sich beherrscht und vernünftig eingerichtet hat. Das ist etwas, was man definitiv vermeiden könnte.

Frage: Es gibt ja aktuell einige Ansätze in der PHP-Welt, wie das PHP-Maven Projekt, was derzeit vorangetrieben wird. Würdest du sagen das ist der richtige Weg?

Das ist wahrscheinlich die richtige Weg, damit habe ich mich aber noch nicht ausreichen dabei beschäftigt.

Es gab auf dem letzten Hackathon in internet Summit Projekt, Magento mit Composer zu verhängen, so dass man sein Modul über die Bibliothek kann Modman Files automatisch installieren kann, mit ihrer Abhängigkeiten. Aber auch dafür benötigt man wieder Hintergrundwissen zu mehreren Tools, um sie vernünftig einzusetzen.

Thema: Integration von Fremdmodulen

Frage: Du hast vorhin erzählt, dass man sich beim Einsatz von Fremdmodulen zunächst die Konfiguration anschaut, um zu sehen was für Rewrites vorgenommen werden und so weiter. Gibt es denn generell Probleme bei der Integration von verschiedenen Modulen aus verschiedenen Quellen oder neuen Modulen in bestehenden Shops?

Ja. In Bezug auf Fremdmodul gehen die Meinungen auseinander, ich sage, wenn man auf ein Fremdmodul verzichten kann, sollte man das tun. So hat man schließlich die volle Kontrolle, wenn man aber doch nicht ohne Aufwand zu 100% weiß, was ein Fremdmodul macht und ob es nicht doch irgendwo in den Konflikt gibt. Seltener man doch in sie in, muss man natürlich auf Konflikt achten.

Es gibt auch Tools, um die Rewrites zu überwachen und Konflikte zu erkennen aber auch um die Möglichkeit überhaupt in Betracht zu ziehen, ist Erfahrung notwendig.

Frage: Hat das nur mit unterschiedlicher Qualität zu tun und dass man nicht sicher sein kann, ob das jeweilige Modul standardkonform ist, oder ist das ein generelles Problem von Magento?

Das ist eher ein generelles Problem. Es gibt aber auch Arten von Modulen, die ich relativ bedenklos einsetzen, dazu gehören Zahlungs- oder Versandarten, weil dies üblicherweise in anderen Modulen und Blöcken überprüften und entsprechen den Standardschnittstellen andocken. Dies ist ausreichen, wenn man nicht noch zusätzlich Kernfunktionalität erweitern will, beispielsweise in der Versandart in der Höhe dazu beistellen zu lassen.

Sobald Modul beistehen Funktionalität anpassen, wird es kritisch, da kann die Code-Qualität noch so gut sein, es ist häufig erforderlich, Klassen zu überprüfen. Auch Modul von wirklich guten Entwicklern wie Vinai Kopp kommen teilweise nicht unter 5 Rewrites aus und wenn man so etwas einsetzen, muss man immer aufpassen.

Frage: Nun hat Magento also an Stellen wie Versand und Bezahlung klare Hot Spots, in die man eigene Funktionalität ohne Konflikte einbaut, hat weiterhin ein Event-Observer-System mit sehr viele Events in die man sich einhängen kann. Wie kommt es, dass trotzdem so viele Rewrites nötig sind?

Es fehlen an zentralen Stellen immer noch Events, beispielsweise bei der Bestellbestätigungsemail. Für jede Anpassung daran muss man das komplette Order-Modul überprüfen.

Es fehlt aber auch bei vielen Entwicklern das nötige tiefergehende Wissen. Zum Beispiel geht die Anpassung von verschiedenen Tabellen und Formularen im Backend üblicherweise ohne Rewrites, nur mit Event/Observer aber viel Wissen es nicht – ich habe selbst erst spät herausgefunden, wie das geht. Dementsprechend haben viele Module noch unnötig viele Rewrites drin.

Thema: Wartung

Frage: Wenn man die oben genannten Richtlinien für das Layout-System nicht einhält, werden Kern-Templates überschrieben und die Update-Sicherheit ist nicht mehr gegeben. Ist das auch ein typisches Problem bei der Einbindung von externen Modulen?

Ja, wobei externe Module Erfahrungsgemäß weniger Problem machen, die bringenden ist die Templates mit und überschreiben nichts. Das Problem sind ganz Systeme, die von Agenturen oder für eine Entwicklung erstellt wurden und in denen das ganze Template-Verzeichnis übernommen wurde. Das ist dann kein Problem von Extensions sondern von demjenigen, der das Magento-Projekt ursprünglich erstellt hat.

Es ist in typischer Anforderung, solche Systeme zu übernehmen, und das ist in undankbarer Aufgabe. Häufig sind fünfstelliger Betrag notwendig, um so etwas wiedergeradezubringen, das hört kein Kunden gerne.

Sonstiges

Thema: Werkzeuge

Frage: Abgesehen vom Deployment, ist die Werkzeugunterstützung für Magento OK?

Debugging, Entwicklungsumgebung, lokale Server... das funktioniert alles mittlerweile sehr gut. Da gibt es eigentlich überall etwas, was out-of-the-box funktioniert.

Es ist aber allem in schon sehr viel besser geworden. Früher gab es zwar auch schon Möglichkeiten, die ich nicht genutzt habe aber seitdem ich PHPStorm und Zend Server CE und den Ding bauten Debugger benutzt habe ich deutlich Vorteil. Das sind relativ kleinen Sachen, die man in einem halben Tag installiert hat, auch wenn man sich vorher nicht damit beschäftigt hat, aber sie machen das Entwicklungsleben schon viel einfacher, auch Kleinigkeiten wie das Magento-Plugin für PHPStorm.

Thema: Einstieg; Erfahrung

Frage: Es ist soweit ich weiß nicht einfach, ein tiefes Wissen über die internen Abläufe zu erlangen. Was sind da die Hindernisse und Schwierigkeiten?

Magento und die Anforderung an die Entwicklung stellt werden sehr häufig unterschätzt, gerade von PHP-Entwicklern. Viel kommen aus dem Bereich und haben vorher andere PHP-Projekte bearbeitet (bei mir waren es Typo3 und xt:Commerce), die wesentlich einfacher weiterentwickelt sind. Magento ist in sehr komplexes System mit Anleihen aus der Softwaretechnik, in das man zum Beispiel als erfahrener C++ oder Java-Entwickler in einem sehr frühen Einstiegs kommt. Für dies ist PHP nur in anderer Programmiersprache mit einigen Besonderheiten, die Mechanismen und Techniken sind jedoch eher an Java u.ä. angelehnt. Eine akademische Ausbildung in der Informatik hilft ebenfalls, allem in Erfahrung mit PHP im objektorientierten Bereich natürlich auch. Mit der „oldstyle PHP Entwicklung“, die man seit 10 Jahren kennt, kommt man dagegen nicht so weit. Ich komme selbst aus dieser Richtung und habe z.B. die ersten zwei, drei Jahre kein Debugger genutzt, weniger Einblick in die verschiedenen Design Patterns gehabt und so weiter.

Als Einstiegspunkt würde ich immer in Schulung von mindestens in einem Tag empfehlen. Die gab es zum Beginn Anfangszeit noch nicht, wodurch der Einstieg deutlich länger gedauert hat als jetzt für jemanden mit meinem Hintergrund dauern würde. Es hilft auch ein gutes Buch dazu zu lesen. Und in Ansprechpartner, dem man Fragen stellen kann, ist sehr viel Wert, ich habe selbst schon in der Entwicklung bei einem Kunden geschult, zunächst zwei Tage am Stück dann projektbegleitend als Ansprechpartner. Er ist dadurch extrem schnell reingekommen.

Frage: Ohne Schulung oder erfahrenen Ansprechpartner sind die Möglichkeiten ja recht gering. Es gibt ein gutes Buch, das aber auch nicht alles abdeckt, es gibt keine offizielle Dokumentation, der Code selbst ist wenig dokumentiert. Muss man sich alles selbst zusammensuchen?

Genau. Mittlerweile gibt es sehr gute Ressourcen im Web, das ist jedoch Stückwerk aus Blogs, Foren und dem Magento-Wiki und nicht an einer Stelle zusammengeschrieben. Aber so in der Entwicklungsdokumentation müsste auch über 1000 Seiten stark sein, die würde sich wahrscheinlich in einem Jahr durchlesen.

Frage: Sie wäre ja nicht nur zum durchlesen sondern auch zum gezielten Nachschlagen hilfreich...

Die Grundlagen muss man sich aber erst mal aneignen, bevor man in die Details gehen kann. Dann findet man auch die richtigen Stichpunkte zum Suchen, wo wir die Kernelmodule aufbaut sind und wo man nachsehen kann, wenn man in der Funktion sucht und wie sie umgesetzt wird. Wenn man sich das selber bringen will, muss man erst mal rein, zwei Module mit Hilfe aus dem Internet basteln und in den Kernel hininschauen. Je mehr man da sieht und erkennt, desto mehr versteht man das ganze Magento-System.

Thema: Weiteres Verbesserung-Potential

Frage: Wo gibt es auf Seiten der Entwickler Verbesserung-Potential?

Mehr auf Qualität achten; die gegebenen Werkzeuge auch nutzen (Debugging, Testing); Schulungsaufwand / Einarbeitungsaufwand in Kauf nehmen und nicht einfach drauf losarbeiten wie man sonst vielleicht im Jahr lang gewohnt ist; auch mal neue Techniken ausprobieren, die in der dann auch weiterbringen; aktuell bleiben; informiert bleiben, darüber was es für neue Entwicklungen gibt.

Frage: Und auf Seiten des Frameworks?

Auf Seiten des Frameworks wäre häufig noch in der Zusammenarbeit mit den Entwicklern notwendig, also dass in offener Entwicklung prozess stattfindet, dass es einfach reiner ist für die Entwickler, Code beitzustellen, z.B. eigene Events hinzuzufügen; dass das ganze etwas flexibler gehandhabt wird. Dazu müsste man natürlich mehr Personal bereitstellen, das für die Communitybetreuung da ist. Mehr Dokumentation auf jeden Fall, aber 2009 war das Jahr der Dokumentation [Anm.: Aussage von Magento Inc. 2008], 2010 wahrscheinlich auch... Da muss man einfach von Anfang an dran bleiben, wie es jetzt bei Magento 2 versucht wird. Da haben sie schon einen guten Ansatz, die sie jetzt mit dem neuen Projekt durchsetzen. Es ist schon viel besser geworden, dadurch dass es Schulungen gibt, dass es Anhandbücher gibt, dass es jetzt Zertifizierung gibt und dass jetzt mit Magento 2 versucht wird, viel transparenter zu gestalten und Contributions zu erleichtern. Aber da ist sich rlich noch Luft nach oben.

Ich hoffe, dass sich bei Magento 2 auch im Vorfeld genügend Entwickler daran beteiligen,

an den passenden Stellen Patch's zu committen um selbst Events einzubringen. Es wird nicht der Fall sein, dass alle notwendigen Stellen Events sind aber das könnte noch deutlich besser sein.

Thema: PHP-Frameworks; Vergleich TYPO3, xt:Commerce
--

Frage: Nach deiner Erfahrung mit anderen PHP-Frameworks: Kannst du Probleme identifizieren, die sich generell auf PHP-Frameworks beziehen? Was von dem besprochenen lässt sich z.B. auf TYPO3 übertragen?

Bei TYPO3 hatte ich grundsätzlich das Problem, dass der Code relativ abgeschottet war, dass die Kernmodule häufig in einer einzigen Technik entwickelt wurden; zumindest hatte ich diesen Eindruck. Das finde ich bei Magento so angenehm, dass die einzelnen Module so aussehen wie die Kernmodule. Bei TYPO3 hat man also deutlich weniger gute Beispiele und Vorgaben. xt:Commerce ist sowohl was andere, weil es eher „gskript“ ist und man deutlich weniger nur Anpassungen am Kernsystem macht, nicht in ständigem Modul.

Frage: Das sind also drei Anwendungen mit sehr unterschiedlicher Architektur.

Magento bietet darunter mit Abstand die besten Grundlagen dafür, hochqualitativ zu entwickeln. Die Ähnlichkeit zwischen den Kernmodulen und einzelnen Modulen ist in sich sehr wichtiger Faktor, also dass man einfach sehen kann, wie ein Modul „gut“ entwickelt ist. Das war bei xt:Commerce überhaupt nicht so und bei TYPO3 nur sehr eingeschränkt.

Frage: Sprich, Entwurf und Implementierung stellen sich sehr unterschiedlich dar aufgrund der verschiedenen Architekturen. Wie ist es mit der Anforderungsanalyse und Spezifikation, lassen sich da die Probleme bei Magento-Projekten auf andere Frameworks übertragen?

Auf xt:Commerce deutlich nicht, weil man sich da z.B. sehr wenig Gedanken um Updates-Sicherheit machen musste, weil es sowohl in der Anfangsphase programmiert hat, war das im Kernsystem, damit konnte man dann je gleich Updates veröffentlichen. Man konnte immer noch externe Module einbinden aber das lief dann immer so ab, dass es in der Beschreibung dazu gibt, in der steht: „In dieser Datei nach dieser Zeile die Zeile einfügen“, in paar einzelnen Dateien vielleicht noch, aber im großen und ganzen wurde das System da zusammengestückt. Das ist mit TYPO3 oder Magento dann doch ganz anders möglich. Von daher war bei xt:Commerce die Aufwandschätzung einfacher, weil man sich keine Gedanken machen musste, wo kann man was kaputt machen, überschränkt man Kerndateien; das sind Sachen, die man in TYPO3 oder Magento vermeiden muss. Auch wenn das anders vielleicht nicht so ernst nehmen, ist es für mich in sich sehr klar, das Qualitätsmerkmal, dass bei den Kerndateien überschränken werden und auch die Klassen nicht über die einfachsten Methoden raus zu werden (z.B. in `pp/code/local` komplette Klassen zu ersetzen), sondern dass man vernünftig Wege der Anpassung findet: am besten über Observer, Source-Moduls, Backend-Moduls oder ähnliches, oder zur Not noch über Reflections, wenn es sich nicht vermeiden lässt. Solche Mechanismen gibt es in TYPO3 vom Prinzip her auch, wobei ich in meinem Wissen nicht ganz so tief gehen wie bei Magento (aber das System kann ich auch nicht ganz so gut). Das hat sehr großen Vorteil, hat aber auch den Nachteil, dass es komplizierter ist, Sachen damit zu entwickeln und es den Einstieg schwierig macht, also den richtigen Ansatzpunkt zu finden.

Thema: Web-Applikationen

Frage: Spielt es eine Rolle, dass Magento ein webbasiertes System ist? Würde

sich der Unterschied im Entwicklungsprozess bemerkbar machen, hätte man ein Client-Server-System?

Ich hab zwar seit Jahren ausschließlich Web-Software entwickelt aber früher hatt ich auch mit anderer Software zu tun und das hat mich in großen Unterschieden.

A.2 Gerrit Pechmann

Datum: 9. Nov. 2012, Dauer: 1,75 Std.

A.2.1 Kurzfragebogen

Nam : Gerrit Pechmann

Firma: aijko GmbH

Position: Geschäftsführer

Entwickler-Team-Größe : bis zu 9 Personen

Magento-Erfahrung: 3 Jahre

Verwendete Prozessmodelle: Scrum für Entwicklung; Kanban für Wartung; Früher: Wasserfall

A.2.2 Transkription

Fabian Schmengler: Welche Erfahrung hast du mit Scrum gemacht?

Gerrit Pechmann: Wir machen jetzt Scrum seit gut anderthalb Jahren. Bis jetzt sind die Erfahrungen größtenteils positiv wenn sich der Kunde wirklich darauf einlässt. Es gibt Kunden, die sagen „finden wir cool, haben wir schonmal gehört“, und es ist ja auch in gewisser Hinsicht um Scrum da. Die sagen „ja finden wir super, weil wir gehört haben, es bringt dem Projekt oder dem Produkt nur Gutes“.

Oft ist bei uns Scrum aber auch verbunden mit Abrechnung über Time & Materials, weil es einfach die für Scrum beste Vertragsart ist. Klar, man kann auch andere Vertragsarten mit Scrum machen, aber für den Kunden ist dann halt nicht so klar, dass wenn er Sprint für Sprint immer mehr faturiert, das dann auch Geld kostet. Das ist so ein Schwieriges: wenn man das dem Kunden nicht zu Beginn des Projekts ausdrücklich sagt oder es ihm nicht so bewusst ist, ist der erstmal glücklich und toll über Scrum, das alles besser macht – was wir dem Kunden aber auch so verkaufen; wir sagen ja auch, es ist besser, macht besser Software – es kostet aber im Endeffekt dann auch Geld, das ist aber ein ganz anderes Problem so in der Kunden/Lieferanten-Beziehung.

Wir haben Kunden, die Shops betreiben und selbst auch Entwickler haben. Mit denen ist es viel einfacher, Scrum zu machen, weil sie selbst auch Erfahrung haben, was Entwicklung bedeutet. Wenn man in der in der Shop-Betreiber als Kunde hat, ist das schwieriger, weil er nur die Kosten sieht, er sieht nicht den Aufwand dahinter, sieht nicht wie die Leute da sitzen und jeden Tag entwickeln und sich den Kopf zerbrechen wie das Marketing des Shop-Betreibers gibt, lösen. Das weiß der Kunde, bei dem ich den Entwickler sitzen, sehr wohl. Mit dem ist es dann auch leichter, Scrum zu machen, weil die intern immer noch sehr unstrukturiert arbeiten.

Wir haben in den Kunden mit großem Webshop, der hat fünf Entwickler und wenn man die Entwickler fragt, wie sie arbeiten, was für Prozess sie haben, kommt die Antwort „Wie, Prozess?“ – es gibt da wirklich keinen Prozess, man schreibt sich die Sachen im Raum zu: Das Marketing schreibt „Wir brauchen in den Button in grün“, dann macht irgendjemand Entwickler, der gerade frei ist, den Button in grün, egal wie lang er braucht. Und wenn er fertig ist, wird

s auf Liv'g schoben, ohne irgend in den Deployment-Prozess, ohne Testing, ohne Testplan. Es wird in Update gemacht, teilweise auf dem Live-Server irgendwas gefixt, dann müssen sie gucken, dass sie irgendwo noch ins SVN bekommen – wenn sie in sie gibt.

Also sie ist teilweise da draußen echt spannend, wenn man sieht, wie auch große Shops im Hintergrund arbeiten. Aber sie funktionieren, die Frage ist halt nur wie lang. Nicht ohne Grund kommen die ja auch auf uns zu und sagen „OK wir haben so viele Probleme, den Shop zu warten, so viele Probleme, neue Features anzukriegen, weil wir immer Angst haben müssen, dass wir irgend in der Stelle was zerbrechen, weil es so unübersichtlich geworden ist.“ Jeder hat auf seine eigene Art in den Spezialfall reingeprogrammiert und kein Vertrauen mehr wirklich an den Code. Das ist so das größte Problem auch von alten Shop-Systemen wie zum Beispiel xt:Commerce. Die sind dann glaube ich auch ganz froh wenn es so einen Ansatz wie bei Magento gibt, wo das ganz erst mal übersichtlich aufgebaut ist, modular aufgebaut ist.

FS: So viel zur Interaktion mit dem Kunden. Wie sieht es mit dem internen Prozess aus, wie hat sich da Scrum bemerkbar gemacht?

GP: Es gibt bei Scrum nicht mehr die Deadlines. Das schließt Scrum vollkommen weg, es gibt also nicht mehr in den Tagen der Kundenbestimmte, an dem das Ding fertig sein soll. Es gibt bei Scrum für den Kunden die Möglichkeit, nach jedem Sprint zu sagen, „OK, das ist in Ordnung, mit dem möchte ich leben.“ Wir lassen uns also bei Scrum nicht mehr auf die Deadlines-Sachen ein, sondern wenn der Kunde leben will, dann geht er leben nach dem Sprint, das ist seine Entscheidung. Und wir müssen intern nur dafür sorgen, dass am Ende des Sprints die Sachen gut bestellt sind, dass alles funktioniert. Das hat bei uns intern schon dazu geführt, dass wir wesentlich ruhiger sind, ich will nicht sagen entspannter, aber sie ist halt in einer Atmosphäre an so in Projekt zu gehen. Du hast gewiss Baustellen, die du in gewisser Zeit auch fertig haben musst.

Das Team committet sich ja bei jedem Sprint auf die Erfüllung bzw. Umsetzung von Stories. Wenn der Product Owner, der mit dem Kunden sitzt, da nicht mitspielt, OK... sie liegt am Product Owner, dass der Sprint erfolgreich wird. Für das Team bedeutet es schon, dass wir mit weniger zeitlichem Druck herumgehen. Schon mit dem Qualitätsdruck, es soll am Ende des Sprints trotzdem perfekt sein, aber halt nicht mit dem Zeitdruck, dass man zum Beispiel so in Latten von Issues hat, die noch nicht fertig sind und in der Woche ist Deadline, dann geht das Ding auf jeden Fall los. Das gibt es nicht mehr, weil das Team weiß wie schnell es voran kommt.

Es gibt die Team-Velocity, die man in Story Points ausdrückt, also jedes Story wird von uns abstrakt in Story Points geschätzt. Dann weiß man, alle zwei Wochen (so lang geht in Sprint) schaffen wir zum Beispiel 25 Story Points, darauf committet sich das Team dann auch. Wenn es weniger schafft, dann werden beim nächsten Sprint erstmal weniger Story Points in den Sprint gezogen, und so verhält sich das dann mit der Zeit sehr gut aus. Man weiß, das Team schafft im Durchschnitt 22 Story Points, die werden erstmal auch geschätzt. Also die Anzahl der User Stories werden in den Sprint gezogen, so dass man sich nicht übernimmt. Man hat in jedem Sprint nur den Fokus auf die Tasks die jetzt gerade wichtig sind. Das ist gut für den Kunden, das ist gut für uns, weil wir uns auf die Aufgaben, die auch schaffbar sind, fokussieren können.

Das ist ja auch etwas, was Scrum macht, also innerhalb des Sprints die Glaskugel über das Team stülpen: das Team ist vollkommen abgeschottet von äußeren Einflüssen, da ist kein,

d r sagt, „Kannst du mal b n? Mach das mal b n schn ll, das muss auf j d n Fall noch r in vor d m Liv -Gang“, das gibt s nicht m hr. Also das T am wird davon abg schott t, da mach n sich and r G dank n drum, also d r Product Own r. D r Scrum Mast r schützt das T am ab r das T am an sich, also di Entwicklung hat durch Scrum unh imlich vi l Z it, sich auf di g rad wichtig n Sach n, ohn di s äuß r n Störinflüss , zu konz ntri r n.

FS: Hilft das ganz Mod ll auch, di Qualität zu rhöh n?

GP: Ich glaub schon, w il man sich rst mal auf w nig Sach n konz ntri rt und w il durch Scrum di s r T am-G dank noch m hr g förd rt wird. In d m Mom nt wo in Bug auftritt muss sich j d r dafür v rantwortlich fühl n, s gibt nicht m hr „W r hat das g macht? Du? Dann musst du das auch fix n“, w il s b i Scrum di s n T am-G dank n gibt, w il das ganz T am sich auf d n Erfolg in s Sprints committ d, das ganz T am sich v rantwortlich fühl t. J d r fühl t sich schl cht, w nn da irg ndwo in F hl r drin ist. Nicht m hr nur d r Front nd-Entwickl r w nn irg ndwi twas im IE7 nicht gut aussieht, da sag n di Back ndl r nicht, „Hi r Front ndl r, was hast du g macht?“ sond rn „Guck mal, hi r ist in F hl r, könn n wir dir irg ndwi h lf n?“.

Es gibt auch di Situation n, dass s in n Block r gibt, das Syst m st ht, da hab n vi l Scrum Tams (wir nicht, da wir r lativ kl in sind) so in Sir n , da drückt man auf in n Button und s müss n sich all inmitt n vom Raum v rsamm ln. W nn das j tzt 50 L ut sind, kommt j d r aus d m Scrum T am für das Proj kt da hin, b rät sich, wi man d n Block r j tzt am b st n lös n kann. Irg nd in r hat di Lösung und in and r r hat vi ll icht noch in n Hinw is und man kommt vi l schn ll r zum Zi l als w nn so in Block r irg ndj mand m zug wi s n wird, d r sich vi ll icht gar nicht damit ausk nnt und rst mal rumfrag n muss. Also das ganz T am fühl t sich für das gut Produkt v rantwortlich. Und das b d ut t für uns halt auch gut Qualität und ich glaub schon, dass Scrum da hilft.

FS: Also abg s h n von d r Sir n macht ihr das schon so?

GP: G nau. Also wir tun dann so, als würd di Sir n g h n und dann komm n halt all zum Scrum Board und sag n „OK, wir hab n hi r in Probl m, wi könn n wir das lös n, woran könnt das li g n?“

FS: Du hast g rad di Story Points rwähnt. Gibt s b i di s r Schätzungstypisch Probl m ?

GP: Also s gibt g n r ll ja vi l Probl m b im Schätz n. Das was man üblich rw is macht, ist so in Exp rt nschätzung: Programmi r r schätz n zum B ispi l Tag , schätz n d n Aufwand in Tag n od r in Stund n. W nn wir noch Tag und Stund n schätz n müss n, dann nutz n wir da auch k in math matisch s Syst m um d n Aufwand zu b rchn n, sond rn wir frag n auch d n Programmi r r „Was schätzt du?“ und d r sagt dann „2 Tag “ od r „2 Stund n“ – das klassisch was d r Programmi r r sagt ist „zw i“, ob s Tag od r Stund n sind.

Und das Schätzprobl m g ht Scrum glaub ich ganz gut an ind m man sagt, wir schätz n nicht di Dau r, wir könn n nur di Größ schätz n. Und w nn d r Kund in n u Anforderung r in gibt und sagt „Ich möcht j tzt di s n Button in Grün hab n, wi aufwändig ist das?“ int r ssi rt s ihm ig ntlich m ist ns gar nicht, ob s j tzt *zwei* Stund n dau rt, d n int r ssi rt nur, sind s h r zw i Stund n, zw i Tag od r zwanzig Tag ? Und di s s „kl in, mitt l, groß“ macht Scrum ganz gut ind m s di s Story Points von 1, 2, 3, 5, 8, 13 hat, nicht durchnumm ri rt von 1-10 sond rn b n in di s n Abständ n. W il j größ r d r Aufwand ist, umso ung nau r ist r. Für d n Product Own r b i Scrum, d r auf Kund ns it sitzt, ist

s dann nur relevant, zu wissen, ist es in eins, zw i od r in fünf („so mitt l“) od r in acht od r in 13, also „ganz groß“. Das richtig ist ns schon, um dann zum Beispiel zu sagen „in 13 für den Button – nein, das ist es mir nicht wert, schick ich erst mal nach hinten“. Also für den Kunden ist es relevant, welche Größen in Anforderung hat, aber abstrakt. Und das macht Scrum ganz gut, dass halt die abstrakten Story Points einführt und es stellt sich dann im Laufe des Projekts heraus, was zum Beispiel in acht zeitlich geschehen bedeutet, auch je nachdem wie das Team zusammengesetzt ist.

FS: Kommt es denn trotzdem zu Fehlinschätzung? Ich habe jetzt oft in Zusammenhang mit Magento gehört und auch erlebt, dass man nicht genau weiß, ist etwas gerade nur in klaren Anpassung oder gibt es widerunwartete Probleme und man muss alles möglich umwerfen. Wie funktioniert das also in der Praxis?

GP: Definitiv kommt es auch bei Scrum zu Problemen, das hat ja nichts mit dem Tool zu tun. Ich will nicht sagen oft, aber es kommt doch ab und an zu Fällen wo man sich komplett überschätzt oder wo man in Anpassung als klar einschätzt aber das System – wenn du sagst Magento implementieren – es ist dann doch nicht so einfach macht, wie man sie eigentlich erwartet hätte. Dass also in klaren Anpassung, zu der man sagt die kann man ganz leicht im Template lösen oder anders lösen, dann doch umständlicher ist.

Wird es auch häufig so ist, dass die einfachste oder schnellste Lösung nicht wirklich die schönste ist. Man kann ja viele Sachen ganz dirty im Template machen und das kriegt man vielleicht nicht hin in der geschätzten Zeit, schafft sich aber ein Problem. Das ist in Problem, was auch Scrum adressiert: die technische Schuld. Also Scrum sagt, „Wenn ihr etwas löst, löst es sauber. Habt keine Angst vorm Refactoring, lasst Refactoring zu, schreibt den Code, den ihr jetzt braucht sauber, gut, gut steht. Habt keine Angst davor, dass ihr ihn vielleicht im nächsten Sprint komplett über den Haufen werfen müsst um die dann beste Lösung zu finden die dann aber sauber und wartbar ist.“ Will man wie schon gesagt sich sonst andere Probleme schafft, die kommen irgendwann zurück.

Wir hatten das jetzt bei in dem Projekt, das seit Januar dieses Jahres läuft, im zweiten oder dritten Sprint haben wir in Anpassung gemacht, die gerade so auf der Kippe zu dirty war. Und jetzt irgendwann in Sprint 23 kommt das wieder zurück. Jetzt merken wir doch, dass die Anpassung die wir da gemacht haben, uns in einen krassen Bug verursacht hat. Die Zeit die wir uns da gespart haben, müssen wir jetzt erstmal investieren um den Bug wieder sauber zu entfernen.

FS: Gut, das ist in typischer Software-Probleme. Ich wollte aber gar nicht so sehr auf Quick and Dirty Lösungen hinaus. Gerade von Magento kenne ich das: es kommt häufig vor, dass man glaubt, es müsste eigentlich in einfach und auch sauber Lösung geben aber dann ist das System gerade anders still nicht flexibel genug, es war nicht vorgesehen, und man muss es dann doch wieder sehr umständlich lösen. Kommt das bei euch so vor?

GP: Definitiv, ja. Ganz oft denkt man, da muss auf jeden Fall ein Event betrachtet werden, um das schön zu lösen, und dann guckt man in den Quellcode und sieht „Oh, Mist. Da ist doch gar kein Event, warum nicht? Das würde so viel Sinn machen, da in Event zu haben!“ Klar, es kommt dann doch vor dass man mal etwas in loc 1/M geablen muss.

FS: Der Aufwand wird wahrscheinlich größer, wenn man das dann ordentlich machen will. Die Frage ist, wie fängt man das auf, dass es solche Fehlinschätzung gibt?

GP: Abstraktionstechnisch oder zeitlich? Zeitlich geschehen ist das mit Scrum kein Problem.

Du hast zum Beispiel die Tasks in zwei Wochen zu erledigen, oder die Anforderungen die du umsetzen sollst. Und die Anforderung dauert deutlich länger als geschätzt. Wir haben gedacht, es ist in der Anpassung in Magento, aber im Endeffekt haben wir dann doch in den zwei Wochen nur die Story geschafft, weil wir in Risikofassungen aufpassen mussten. Das drückt im Prinzip die Änderungen in den Stories die offen sind aus dem Sprint raus. Wir haben dann – leider – nur die Story geschafft und die Änderungen müssen im nächsten Sprint gemacht werden.

Scrum geht hin und macht absichtlich die sogenannten Timeboxing. Alles was man macht, wird irgendwie timeboxt. Ein Sprint zum Beispiel auf zwei Wochen oder in Wochenjahren, das kann man selbst festlegen. Das Daily Standup was man jeden Tag macht wird timeboxt auf 15 Minuten. Also egal was passiert, nach 15 Minuten gehen alle weg. Nicht, „komm lass noch mal reden darüber reden“, nein, ist nicht. Und das gleiche gilt halt bei im Sprint: Wenn nach zwei Wochen nur die Story geschafft wird, weil wir uns vollkommen verschätzt haben, ist das halt so.

Das ist dann eher ein Problem, dem Kunden zu sagen, guck mal hier, wird doch wahrscheinlich teuer, und da kommt dann wieder das Vertragswerk ins Spiel, das man halt vorher gut festzulegen sollte, weil so etwas halt passiert, das ist so in der Softwareentwicklung. Das kann man nicht immer ausschließen.

FS: Es wird also auch nicht mitten im Sprint gesagt, wir setzen die Story Points jetzt bisschen mal hoch.

GP: Genau das ist es. Wenn wir mal bei dem Beispiel bleiben, du hast die Stories, die mit eins geschätzt, die mit drei, die mit fünf, wir haben also neun Story Points in diesem Sprint. Als erstes soll die Eins erledigt werden, die Eins verschlingt jetzt komplett den ganzen Sprint. Dann haben wir nur die Story Point geschafft. Das heißt aber auch, dass wir im nächsten Sprint nur die Story Point schätzen. Das ist jetzt ein bisschen konstruiert aber sagen wir, wir schaffen die Eins und die Drei, und die Fünf fällt raus, dann werden wir bei im nächsten Sprint erstmal nur vier Story Points reinnehmen, werden dann aber feststellen, dass wir doch mehr geschafft haben und so nähert sich zu vier und zu wenig geschätzt an und man bekommt im Laufe des Scrum-Projekts die Velocity vom Team, die realistisch ist raus.

FS: Die Eins bleibt auch im nächsten Sprint ein Eins?

GP: Die Eins bleibt immer ein Eins. Wir haben sie mit eins geschätzt und machen sie fertig. Dass sie im Endeffekt doch die ganze Zeit gebraucht hat, solche Schätzfehler bügeln Scrum im Laufe des Projekts aus und nähert sich der wahren Velocity an.

FS: Wenn man mit einem Task beginnt, wird man ja erstmal irgendwie in den Entwurf machen. Was gibt es da für Probleme, oder was ist da vielleicht spezifisch für Magento- und PHP-Projekt?

GP: Wir bekommen die Anforderungen vom Kunden, analysieren die erstmal, gucken was wir dafür anpassen müssen... Magento gibt ja schon gewisse Patterns vor, die man nutzen kann, um Probleme zu lösen. Wir müssen gucken, was die sauberste Möglichkeit ist. Wir versuchen immer erst die sauberste Möglichkeit zu finden, in gewisser Weise schränkt uns das Magento schon ein, weil wir da nicht alle Möglichkeiten haben, die man sich so vorstellt. Es gibt andere Frameworks wie Flow3, wo es Aspekte gibt (Aspektorientierte Programmierung), da kann man Anpassungen reinbringen ohne irgendwie in das System eingreifen zu müssen. Nicht so wie das Event-System von Magento aber die Events helfen in dem auch schon beim Entwurf

von Lösung n von Anford rung n. Man w iß, an d r St ll kann man sich inhäng n, um di s Anford rung n in di Standardinstallation inzubring n, ohn vil s lbst zu programmi r n. Di Sach n, di Mag nto li f rt, h lf n in m schon, d nk ich, in n ord ntlich n Entwurf zu mach n.

FS: Ind m man di Patt rns nutzt, di Mag nto b r itst llt od r s lb r nutzt. Es gibt sow it ich w iß k in offizi ll n Guid lin s dazu, man muss also s h n wi s im K rn g handhbt wird. Also Guid lin s, wi man Probl m löst, auf Entwurfs-Eb n .

GP: Das stimmt, ja. Also da hilft s vil, zu guck n, wi Mag nto s lbst an Probl m h ran g ht. G rad w nn man sich das Ent rpris Packag anguckt, was ja auch auf d m s lb n Co r aufstzt, di s m M ge Packag : G rad da si ht man häufig, wi di Ent rpris -Entwickl r Probl m lös n musst n, da si auch di Anford rung n hab n, nicht d n Mag nto-K rn anzufass n um di Ent rpris -F atur s hin inzubring n. Und da si ht man schon, w leh Mög lichk it n s so gibt.

FS: Habt ihr da irg nd twas h rausg arb it t od r k nnt ihr all d n K rn und di Ent rpris Edition infach gut g nug?

GP: Das wär ig ntlich mal in Id , das stimmt. Ein ig n Probl mlösungs-S it im Wiki mal anzul g n. Stimmt, hab n wir nicht... ab r gut Id .

FS: OK, w nn man j tzt in ganz Softwar s lbst ntwick ln würd , würd man d n Entwurf mit Klass ndiagramm n, S qu nzdiagramm n und so w it r noti r n. Sind das W rkz ug , di hi r noch Sinn mach n?

GP: Also um hrlich zu s in wirst du b i uns kaum Klass ndiagramm s h n. In d r Th ori ist das imm r all s ganz toll, in d r Praxis braucht man das kaum und s b zahlt in m auch kaum j mand. W nn wir kompl x Modul schr ib n, hab n wir das schon mal g macht um rst mal d n Üb rblick zu hab n, w leh Akt ur gibt s j tzt in d m ganz n Spi l. Das macht schon Sinn. B i in m Flow3-Proj kt hab n wir s mal so g macht, w il das ähnlich wi das Z nd Fram work kaum Vorgab n macht und da musst n wir uns auch in n Üb rblick v rschaff n. Da wir da auch Tam-üb rgr if nd g arb it t hab n, gab s im Wiki in Ansicht, wo all Klass ndiagramm abg bild t war n. Ab r g rad b i Anpassung n für Mag nto und b i Modul n für Kund n, also w nn wir nicht ig n Modul schr ib n, dann mach n wir das kaum.

FS: Es ist ja auch so, dass di m ist n Akt ur T il d s Fram works sind, man sich in di Hot Spots inhängt und s zwisch n ig n n Klass n h r w nig Int raktion gibt.

GP: Das ist richtig.

FS: St llt s in V rständnis-Probl m dar, dass man di s n Üb rblick vil icht gar nicht so hat, da vil im Fram work passi rt? W iß man trotz d m, was da passi rt?

GP: Mag nto hat ja in zi mlich unüb rschaubar Anzahl an Klass n und Obj kt n, di da zusamm nhäng n ab r w nn man g nug Erfahrung hat und w iß, was d r Standard macht, v rli rt man nicht so schn ll d n Üb rblick. Ab r s b st ht auf j d n Fall G fahr, w il das Fram work an sich (ich s h Mag nto auch h r als Fram work) schon s hr kompl x und mächtig g word n ist üb r di Jahr . Und s st llt in n doch imm r wi d r vor H rausford rung n, da d n Üb rblick zu b halt n. Ich könnt s mir zum B ispi l nicht vorst ll n – und ich glaub vil Mag nto-Entwickl r müss n das noch so mach n und mach n das so –

ohn Debugger auszukommen. Weil das einfach zu viel Sachen passieren, zu viel mit Events gearbeitet wird als dass man das den Überblick behält, was gerade was macht.

Ich hatte heute morgen noch den Fall, dass ich mich gefragt habe, warum der Indexgerade neu aufgebaut wurde. Ich habe in der Indexverwaltung gesehen, dass der Index vor in der Virtualisierung neu aufgebaut wurde, habe nachgeschaut, was zu dem Zeitpassiert ist. Dann habe ich gesehen, dass in der Bestimmungsgangcancel wurde. Und bei in der Gangcancel der Bestimmung wird der Stock wieder ins hochgestzt, und das hatte zur Folge, dass neu indiziert wurde. Ganz viel Sachen, die überhaupt nicht linear laufen und wo man einfach mit dem Debugger gucken muss, was passiert.

FS: Interessante Thema, „unwartete Situationen“ – die treten ja auch und gerade bei der Integration von verschiedenen Modulen auf, wo man nicht genau weiß, was passiert im Hintergrund, wo könnte das noch Einfluss haben... erzähl doch mal etwas zur Integration.

GP: Spannend, auf jeden Fall. Gerade was die Qualität von Fremd-Extensions angeht, da ist man immer wieder erstaunt, dass Leute überhaupt Geld dafür verlangen, was sie da verkaufen. Das ist sogar manchmal erschreckend. Wir haben oft schon den Entwicklern – und das sind ja teilweise Firmen, Agenturen, die selbst auch implementieren und integrieren – E-Mails geschrieben mit Verbesserungsvorschlägen, die sie vielleicht in ihrer neuen Version einbauen könnten. Weil sich auch überhaupt nicht an irgendwelche Coding Guidelines gehalten wird, und es gibt ja irgendwo zumindest Guidelines dafür, wo der Code aussuchen soll. Und wenn vollkommen davon abgesehen wird, in allem was nur geht, fragt man sich, ob es nicht doch sinnvoll gewesen wäre, das ganz selbst zu schreiben. Der Aufwand, den dies kaputt Code mitbringt, der zwar funktioniert aber überhaupt nicht mehr wartbar ist, ist letztendlich groß. Und man hat ja auch die Verantwortung dafür, dass der Code auch in zwei Jahren oder nach einem Update noch funktioniert, das ist halt nicht gegeben bei solchen Fremd-Extensions. Und das ist die große Herausforderung bei Integrationsgeschichte von Fremd-Extensions in Magento: zu gucken wie sie aufgebaut sind und ob es Konflikte gibt. Denn die verschiedenen Systeme ist ja auch zufällig, wenn also zwei Extensions in Klassen überschreiben, haben wir in der Problematik. Das ist auf jeden Fall eine große Herausforderung, vor der man gestillt wird, wenn man sich fremde Code ins System holt.

FS: Jetzt sprichst du das Re-write-System und Konfliktpotential an. Es ist auch bei richtig guten Extensions der Fall, dass Konflikte auftreten oder sie nicht zusammen funktionieren. Wie geht ihr da heran, die Konflikte zu entdecken und vielleicht zu beheben?

GP: Man kann sich da im Prinzip verhalten, indem man versucht... das funktioniert in gewisser Weise schon, teilweise, wenn die Re-writes so aufgebaut sind, dass sie wenig verändern. Wenn Klassen komplett umgeschrieben wurden, so dass da gar nichts mehr ist wie vorher, dann nutzen wir die Extensions auch einfach nicht. Wir achten schon darauf, dass wir uns nicht allzu hohes Risiko ins Projekt holen. Im Zweifel lieber nicht und wir uns lieber gegen die Extension, auch wenn wir sie gekauft haben. Dann müssen wir sie eben zurückgeben und lösen das ganze lieber selbst, sauber. Wir versuchen da auch nicht zu viel Workarounds zu machen, sondern im Zweifel lieber das ganze rauszukicken und neu zu machen.

FS: Es gibt aber schon Workarounds, habe ich das richtig verstanden? Also wenn es nur ein punktueller Änderung gibt, die sie in einem eigenen Modul zusammenzuführen?

GP: Ja, man kann auch hingehen und in der neuen Extension die Klasse von der anderen Extension erben lassen. Eine Klasse darf also nicht von der Basis-Klasse erben, sondern von

der der anderen Extension. Das Problem ist dabei, dass man sich die Upgrade-Fähigkeit der Extensionen wieder kaputt macht. Das sind alles Workarounds, wo es immer in der Tradition gibt und das ist halt unschön. Es muss irgendwo dokumentiert sein, bei einem Update fliegt in man sonst alles um die Ohren. Man schafft sich nur Abhängigkeiten.

Zum Thema Integration: Wir arbeiten mit Modman, weil es aus unserer Sicht schon ein großes Problem ist, dass ein Modul total über die Kernel-Anwendung verstrut ist. Eigentlich wollen wir ja, dass Modul das System in kleinen Brücken rewirrt. Und was wir machen ist, wir trennen die Kernel komplett von der Erweiterung, haben das auch physisch an unterschiedlichen Orten und fügen das dann im Grundbündel zusammen. Erst dann verpacken wir wieder die Kernel und die Module und haben so inhaltlich besser Upgrade-Fähigkeit was die Kernel angeht und die Module, weil wir die Module auch in einem externen System haben können. Das sind dann oftmals Submodule aus Git oder Extensions aus Subversion. Und so können wir ganz einfach in Extension oder die Kernel updaten ohne die Verpackung aufbrechen zu müssen.

FS: Das führt uns auch schon zum Testen: Testet ihr die Module auch isoliert, also auf einem eigenen „sauberen“ System oder rüstet alles zusammen?

GP: Wir testen das zusammengefügt. Also wir testen in Extension nicht zu rüstet auf einem frischen Magento, sondern testen direkt im Projekt. Ich muss ehrlich sagen, wir nutzen auch nicht allzu viele Fremd-Extensionen, wir versuchen immer alles zunächst auf unsere Art zu lösen, wenn es nicht zu aufwändig ist. Wenn das ein ganz krasses Anforderung ist und es da schon ein gutes Extension gibt, die man für 500 € oder 150 € kaufen kann, dann gucken wir die an, probieren die. Und das hatten wir auch bei dem Plus was in Anbindung an Preisvergleichsmaschinen macht. Im Endeffekt haben wir da aber so viel Anpassungsaufwand in die Extension gesteckt um sie wieder aufzuräumen, da hätten wir sie auch von Anfang an selbst schreiben können. Also wir haben, muss ich sagen, wirklich schlechte Erfahrung mit Fremd-Extensionen gemacht. Das ist schon ein großes Problem, dass die Qualität überhaupt nicht stimmt, die man sich da holt.

Deshalb gehen wir sofort im Projekt, natürlich nicht auf dem Live-Server oder Staging-Server, jeder Entwickler hat seine Umgebung. Wir haben in der großen Server, auf dem ganz viele virtuelle Maschinen laufen und jeder Entwickler verbindet sich auf seine VM und entwickelt da, testet da. Er guckt sich dann den Code von seiner Extension an, die man sich reinholt oder selbst schreibt und guckt, ob sie funktioniert, macht also Integrationstests. Wenn das funktioniert, geht es weiter.

FS: Nutzt ihr Unit Tests?

GP: Nein.

FS: Warum nicht?

GP: Das ist eine gute Frage. Für eine Extension, die wir gerade selbst schreiben, schreiben wir auch Unit Tests. Es ist ein schwieriges Thema. Wir wollen eigentlich alle Unit-Testen aber es ist halt doch erheblicher Aufwand. Wir haben dabei in dem Typo3-Projekt gemacht, da hat der Kunde selbst bezahlt, und da haben wir uns auch sehr geschätzt. Es ging glaube ich um 10 Tage Entwicklung, wir haben 5 Tage für Unit Tests geschätzt, haben im Endeffekt initial für die Unit Tests 7 oder 8 Tage gebraucht. Man überschätzt sich dabei nicht, was es für einen Aufwand bedeutet, jeder Klugheit, jeder Unit zu testen.

FS: Vielleicht ist das auch in Problem, die Tests separat zu betrachten und zu schätzen? Habt ihr da in den Test First Ansatz verfolgt und alle Tests zu schreiben?

GP: Ja, also wir haben wirklich getestet entwickelt. Ich bin ja kein Typo3ler aber ich habe auch mit dem Typo3ler gesprochen, der das macht – das war unser erstes Projekt wo wir das durchgezogen haben, das sollte dann auch 100% sein – und er hat gesagt, das ist nicht ohne. Und das in dem Kunden zu verkaufen ist unheimlich schwer. Das war in großem Agitur, die haben gesagt, „auf jeden Fall, das ist in Risikokunden von uns, finden wir gut, dass ihr das vorschlagt, macht mal, das Budget ist da.“ Aber das in dem Mittelständler zu verkaufen, ist glaube ich nicht so einfach.

FS: Hast du in den Vergleich, wie Unit Tests mit in dem Anwendungs-Framework wie Typo3 zusammenarbeiten, im Vergleich zu wirklichigen Anwendung?

GP: Nein, da habe ich keine Erfahrungswerte. Ich weiß, wie Submagnto funktioniert. Eine Extension haben wir jetzt angefangen, mit Unit Tests zu versehen, das stellt uns aber auch noch vor in paar Herausforderungen, gerade was Controller und so etwas angeht. Normalerweise, OK, das ist relativ einfach zu testen, aber in 100% Abdeckung zu haben.. da arbeiten wir noch dran.

FS: Das heißt, ihr habt auch mit in dem der Magnto Unit Test Frameworks gearbeitet?

GP: Nein, direkt PHPUnit zusammen mit eigenen Klassen.

FS: Wie handhabt ihr dann die vielen Abhängigkeiten zu Magnto, die man meistens noch hat? Da müssen ja Mocks geschrieben werden und Fixtures, und das kann recht kompliziert werden.

GP: Also das hat uns noch nicht vor so in Herausforderung gestellt weil unsere Extension auch gar nicht so viel mit Magnto direkt zusammenarbeitet. Der Anteil an Objekten, die direkt aus Magnto kommen ist relativ gering. Deswegen war das nicht so das Problem.

Das war die Extension, die Magnto mit Typo3 verbindet und der Anteil, der da auf Magnto-Site passiert, also mit den Magnto-Objekten wie den CMS Pages, ist vernachlässigbar. Deswegen war es relativ einfach zu gucken, wenn ich das und das eingebaut kommt die CMS Site mit folgenden Daten zurück.

FS: Es gibt also Typen von Units, die einfach zu testen sind, weil sie sehr wenig Interaktion mit dem Framework haben und da würdest du sie auch empfehlen?

GP: Ja, ich glaube das hat uns auch PHPStorm gut unterstützt, denn das kann relativ gut so Sachen initialisieren. Ich glaube deswegen sind wir auch um die Magnto Unit Test Sachen herum gekommen, weil wir so bei jedem Test das ganze Framework initialisieren, und dann Zugriff auf die zu testenden Objekte haben. Deshalb war das nicht so das Problem.

FS: Was für Werkzeuge nutzt ihr für funktionale Tests?

GP: Momentan ist das alles manuell bei uns. Gerade wenn man agil arbeitet, wäre es natürlich super-schön alles automatisch zu haben, weil das irgendwann auch zum Pflichtprogramm gehört. Da macht es uns Magnto aber nicht einfach. Aus den Gründen dass das Framework von Haus aus eben nicht vollständig Unit-testbar ist, im Gegensatz zu Oxid zum Beispiel

Funktionale Tests, darunter würden ja auch Selenium Tests fallen. So etwas machen wir momentan noch nicht, sind jetzt aber auf dem Weg dahin. Da muss auch das Projekt dafür

das in und der Kunde die Notwendigkeit sehen. Und beim in großen Projekt, was momentan in der Finalisierungsphase ist, wird es auch kommen. Da streben wir das auch an, dass wir die wichtigsten Komponenten zumindest von dem Magento-Shop funktional stellen.

FS: Es hat also schon noch der Kunde Einfluss darauf, je nachdem ob es jetzt schon hat oder nicht? Es ist nicht der Gedanke, dass ihr vielleicht auch effizienter entwickeln könntet mit Teams?

GP: Es fällt bei diesem Projekt jetzt so langsam auf, weil das hier sehr komplexe Komponenten drin sind, und wir empfehlen das dem Kunden. Es ist halt trotzdem ein Problem, wenn der Kunde es nicht zahlt, das dann auf die eigenen Kappen zurückzuführen. Wir würden, da ich dich vollkommenerweise nicht, vielleicht ein bisschen arbeiten aber so lang der Kunde das nicht versteht...

Der Kunde spart zwar im Endeffekt, hat weniger Fehler, muss weniger für Bugfix bezahlen, sein Entwickler haben auch mehr Spaß. Trotzdem, wenn es nicht insicht... da sind wir dann doch Dienstleister und wenn der Kunde es nicht will, oder es nicht versteht: das ist noch das Problem, wenn der Kunde die Notwendigkeit nicht sieht und sagt, „Egal, bevor du jetzt Teams schreibst, die ich nicht versteht, zahle ich lieber für Bugfixing.“ Das ist aus Software-Entwickler-Sicht überhaupt nicht nachvollziehbar aber für den Kunden... ich weiß nicht, da steckt man manchmal nicht drin.

FS: Seid ihr in dem Zusammenhang schon mit dem Magento TAF in Betrachtung gekommen, den SLSium-Teams, die den Core abdecken soll?

GP: Habe ich gesehen, wir haben es noch nicht eingestzt. Das wäre mal spannend, das zu machen. Ich glaube, am Anfang war bei Magento alles noch neu und die Shop-Beitrag war natürlich drauf, Hypothese und so weiter, ich glaube dass es langsam Zeit wird, da so ein gewisse Professionalität reinzubringen, das sollte einfach dazugehören. Ich glaube, in Zukunft wird man in den Shop ohne Testing, ohne automatisierte Tests vielleicht gar nicht mehr verkaufen, weil man sagt, das ist einfach immer drin: „Wenn ihr in der dirty Lösung wollt, sucht euch jemand anderen aber von uns erwartet ihr Qualität und die können wir euch nur liefern, wenn wir die Teams als Teil des Standard-Pakets ansehen.“ An dem Stand sind wir noch nicht aber ich glaube schon, dass wir dahin gehen müssen, zu sagen, ihr wollt gute Software und ihr wollt damit verkaufen, das gehört einfach dazu. Nicht ohne Grund kosten die großen Shop-Systeme zehntausend an Lizenzgebühren, es sollte einfach dazu gehören. Ich glaube, dass die Branche, also die Dienstleister, weg müssen von diesem Hinterrhof-Image. Man hat ja als Magento-Entwickler immer noch genug Anruf von anderen Agenturen, die sagen, „Wir haben hier in den Shop für 3000€ in dem Kunden verkauft, aber irgendwie funktioniert das nicht, könnt ihr mal gucken?“ Das sind die Probleme, die man hat, die auch die kleinen Agenturen haben, die total überfordert sind von der Komplexität dieser Software.

FS: und da dann nachträglich die Qualität wieder zu steigen ist schwer.

GP: norm schwer.

FS: Euer Projekt macht ihr aber hier von Anfang an, oder habt ihr auch solche Übungen von „missraten“ Shops?

GP: Nein, also üblicherweise machen wir die Shops neu. Es sind halt so kleine Sachen aber und an, wo dann Hilfe benötigt wird. Die macht man nicht immer gerne, weil es schon in undankbarer Aufgabe ist, aus so einem verhassten Shop etwas sinnvolls zu machen. Wenn man da nicht unbedingt viel Aufwand reinsteckt, bleibt immer irgendwo noch ein missraten

Stell und das fällt dann doch auf ihn zurück. Die sag n dann, „Ihr habt den Shop doch gefixt und da ist trotzdem noch in Fehler.“ Ja, man kann nicht alle Fehler finden, die da irgendwo in programmiert wurden oder die vielleicht sogar schon im Code sind. Fehler gehen halt irgendwo doch zur Softwareentwicklung dazu, man wird kein 100prozentig fehlerfrei Software herstellen außer vielleicht man schreibt Software für Flugzeug-Cockpits.

FS: Kannst du nochmal einen Überblick über den Qualitätssicherungs-Prozess geben? Wir haben uns jetzt einige Schritte im Hinblick auf die anschaut aber wir laufen zum Beispiel die manuelle Tests ab?

GP: Wenn in Entwicklung ein Bug gefixt hat oder in unserer Feature implementiert hat, geht das in die Qualitätssicherung. Vom Prozess her ist das so, dass er physisch am Board die Karte auf „QA request“ setzt. Dann geht in andere Entwicklung für das Vire-Augen-Prinzip hin, und schließt auf „in QA“, das heißt, er stellt sich jetzt gerade. Und wie gesagt wird, da haben wir keine definierten Vorgaben. Er guckt sich den Code an, der committet wurde, er stellt die Funktionalität wieder vom Entwickler programmiert hat beschrieben und vom Entwickler den Bug oder das Feature erfasst hat, prüft ob die Anforderungen erfüllt sind. Wenn es in Feature ist, schreibt der Entwickler das implementiert hat auch in die Anleitung, was getestet werden soll, wie man auch die Eckfälle testen kann und der Tester prüft dann, ob das alles so funktioniert.

Zusätzlich haben wir in den Code-Review-Prozess, der asynchron läuft, also wenn der Entwickler dann von der Funktion her getestet hat, sagt, es funktioniert, ist das Ticket „done“. Asynchron ruft der Entwickler den Fehler gefixt hat, oder das Feature implementiert hat, in den Code-Review und lädt dazu alle Entwickler ein, die mit dem Projekt was zu tun haben, möglichst auch Entwickler, die vom Skill her über ihm sind. Er möchte ja auch weiterkommen und lernen, da ist so in den Code-Review-Prozess ganz hilfreich, weil sich das Team so verbessert und hochzieht. Bei dem Code-Review wird überhaupt keine Funktion getestet, sondern lediglich Code-Qualität, Coding Guidelines, der Stil ein bisschen.

Das ultimative Ziel ist ja, egal wie der Code geschrieben hat, guckt man drauf und sagt, „ja das war ich vor ein Jahr“, dabei war man das gar nicht, sondern jemand anders im Team. Dass man sich vom Team her so annähert, dass alle gleich Code schreiben, jeder kann den Code sofort lesen, dann hätte es auch selbst sein können, der es geschrieben hat. Das ist bei uns schon ein wichtiger Prozess, da man in diesem Code-Review Abschnitt auch nochmal Fehler im Code findet, die so vtl. nicht offensichtlich sind (zum Beispiel die klassische if-Bedingung mit nur einem Gleichzeichen. Kann super funktionieren, der Ausdruck ist halt true, und so etwas fällt dann beim Code-Review erst auf).

FS: Das Code-Review wird tatsächlich für jeden Task im Team mit dem ganzen Team durchgeführt

GP: Ja, genau genommen für jeden Commit. Also immer dann wenn etwas committet wird, bzw. bei Git ins zentrale Repository gepusht wird, wird dazu in den Code-Review angelegt. Das wird auch mit dem Ticket verbunden, dazu ist die Software von Atlassian ganz gut. Wir haben schon Tools, die miteinander kommunizieren können, dass man auch am Ticket sieht, was dazu committet worden ist. Also die Abhängigkeit zwischen der Aufgabe und dem was letztlich im Repository landet, ist sichtbar für alle.

FS: Was ist das für ein Tool, was die ständige Tracing zwischen Commit und Tasks macht?

GP: [Demonstration] Wir nutzen Jira für die Tickets und wenn man sich jetzt so in Ticket

anguckt, ist das mit uns r m Git Repository v rknüpft. Das ist auch in Softwar von Atlassian, n nnt sich Stash und ist im Prinzip Github hint r d r Fir wall, mit ähnlich n F atur s. Hi r si ht man auch di Branch s. Uns r F atur s ntwick ln wir in F atur -Branch s, g -tr nnt vom aktu ll n Entwicklungsstand und w nn si f rtig sind, w rd n si g m rg d in „d v lop d“. Di Commits w rd n also b im Tick t aufg list t, und umg k hrt. Ich s h di Issu s, di mit in m Commit g löst sind als V rknüpfung b im Commit. Das wird anhand d r Commit-Nachricht rkannt, man muss dazu di Tick tnumm r in di Commit-Nachricht schr ib n. Vom Tick t aus g ht s zurück und ich s h , w leh Dat i n g änd rt wurd n.

Ähnlich für R vi ws: Crucibl , auch von Atlassian, crawlt di ganz Zit das Repository und st llt di Commits grafisch dar. B i di s r Softwar gibt s auch di Möglichk it, R vi ws zu rz ug n. Ich s h dann, was in d m R vi w g änd rt wurd und w r daran t ilg nomm n hat.

FS: B im R vi w w rd n also dir kt Änd rung n vorg nomm n?

GP: Genau, hi r schau n sich L ut d n Cod an, schr ib n dann Komm ntar (das muss nicht imm r ganz rnst s in) und n hm n Änd rung n vor. Das m int ich vorhin auch mit asynchron, wir s tz n uns nicht für Cod R vi ws zusamm n, sond rn mach n das üb r das Tool: rz ug n das R vi w, lad n di L ut in, di sich das anguck n soll n und di könn n dann Komm ntar r inschr ib n und auch schon klassifi z r n, F hl r sind rot, Komm ntar g lb. So Sach n wi „Klass n-Komm ntar f hlt.“. Das fällt vi ll icht b im Programmi r n und auch b im funktional n T st n nicht auf, d nn s funktioni rt ja all s. Es fällt ab r dann auf, w nn man sich d n Cod anguckt. [End d r D monstration]

FS: Ab r so twas wi in n Cod Sniff r, d r b im Commit automatisch di Guid lin s od r Komm ntar üb rprüft, gibt s nicht?

GP: Gibt s nicht.

FS: Hat das in n Grund? Habt ihr uch da b wusst g g n ntschi d n?

GP: Nein, das hat k in n Grund. Wir hab n s bish r infach zitlich nicht g schafft, so twas aufzubaun. Wir hab n für uns r „Lat st“ Syst m , di imm r wi d r n u g baut w rd n (Lat st Build), in n J nkins aufg s tzt, das ist ab r noch in d r T st-Phas . Darüb r würd n wir dann b n auch di ganz n Cod Sniff r lauf n lass n und w nn s S l nium-T sts gibt, di damit trigg rn. Das ist in d r Th ori m ist all s p rft, in d r Praxis noch nicht.

FS: Du sagt st, di Entwickl r schr ib n in di Sp zifikation b r its Hinw is zum T st n. Wi g ht das vonstatt n, im Nachhin in w nn man f rtig ist od r kommt das schon mit d n Anford rung n zustand ?

GP: Das hängt davon ab... also w nn wir nicht Scrum mach n, sond rn im Wartungsv rtrag sind, w rd n di Anford rung n infach als Aufgab n r ing g b n. W nn wir Scrum mach n, komm n di Anford rung n in Form von Us r Stori s. Und das h ift, d r Kund b schr ibt aus Sicht d s Us rs, was g macht w rd n soll und wir ntsch id n, wi wir s t chnisch ums tz n. Solang d r Kund uns nicht irg ndw leh t chnisch n Vorgab n macht, lös n wir s irg ndwi t chnisch saub r. Das h ift, di inzig Anford rung di wir hab n ist, was passi r n soll.

FS: Ab r ihr schr ibt dazu nicht vorh r in Sp zifikation.

GP: Wir lös n s dann, w nn di Anford rung n in in n Sprint komm n. W nn d r Product Own r also sagt, im nächst n Sprint möcht ich di und di Anford rung n ums tz n, st ll n

wir uns hin und diskutieren gemeinsam, wie wir das lösen und erstellen dazuentsprechend Aufgaben. Also der Backendler sagt, ich muss dafür Modell und Controller schreiben, also in der Extension, macht sich die Aufgabe „Extension schreiben“. Man diskutiert darüber, aber es gibt kein festgeschriebenes Spezifikation, was aus unserer Sicht aber auch so gewollt ist. Wir wollen kein definiertes Spezifikation, sondern in diskutieren, global, zusammenentwickelt.

FS: Aber es gibt definiertes geschrieben Hinweis, was getstet werden muss, ggf. mit Sonderfällen?

GP: Ja, genau. Also es gibt sogenannte Akzeptanzkriterien. Wenn die User Story zum Beispiel ist, „Ich möchte mich als Kundenkonto inloggen und für den Kunden Bestellung ausführen können“, steht überhaupt nicht darin, wie wir das technisch lösen. Aber es gibt zum Beispiel in Akzeptanzkriterium, dass man dies sicher nur mit einer IP vom Kunden nutzen kann. Also immer nur dann, wenn der Kundenkonto sich im Gebäud mit dieser IP vom Kunden auf dem Shop verbindet, soll der das sicher überhaupt nutzen können. Das sind dann technische Akzeptanzkriterien, die wir dann auch testen müssen und als Vorgaben haben.

FS: An welchem Punkt werden die aufgestellt?

GP: Zu Beginn, also das ist in der User Story. Entweder schreibt der Kunde sich von sich aus oder es ergibt sich im Gespräch. Also wenn der Kunde dem Team mitteilt, er möchte nächste Woche die User Story umsetzen, fragt das Team, „Welche Einschränkung gibt es, sollen wir vielleicht darauf achten, sollen wir in IP-Sperre einbauen?“ Dann sagt der Product Owner, „Ja IP-Sperre muss drin sein und das und das muss auf jeden Fall funktionieren“, dann schreiben wir das hinzu.

Scrum fördert die Kommunikation, egal ob das mit dem Kunden ist oder untereinander. Über diese Sachen wird gesprochen und dann schreiben wir das schon irgendwo nieder. Aber wir schreiben eben nicht, wie wir das lösen, wir schreiben also eben kein Klassendiagramm, oder Ablaufdiagramm, sondern schreiben nieder, worauf wir achten sollen, was ist die Lösung.

FS: Noch mal zu Situationen an und erwarteten Stellen. Testet ihr die auch in der Qualitätssicherungsschritt oder inzwischen woanders?

GP: Da sprichst du in schwierige Themen an. Es ist tatsächlich so, dass am Ende von diesen Umsetzungen im Prinzip alles nochmal getstet werden muss, und das ist momentan in der sehr große Herausforderung für uns, weil wir kein automatisiertes Testen haben. Deswegen stehen sich auch in den nächsten Monaten in der ganz großen Fokus darauf, dies einzuführen. Denn das ist das größte Problem, der größte Punkt bei uns, dass Situationen, die durch Anpassungen passieren, für uns mit der Zeit unkontrollierbar werden.

Wir haben jetzt in der Fall, wo wir für die Produktkonfiguration in der JavaScript-Monsterbauern mussten. Stell dir bei Magento die konfigurierbaren Produkte vor, wo teilweise 500 Produkte dahinterstecken, mit Abhängigkeiten und bestimmten Konfigurationen, die nicht miteinander funktionieren. Das wurde im Frontend so aufgemotzt, dass das alles mit JavaScript für den User sichtbar gemacht wurde. Und wir kommen nicht drumherum, dies JavaScript zu Unit-testen und dies Site mit Selenium zu testen, die man dann nachher in der Katgori-Ansicht sieht. Dies ganz sicher soll nämlich auch auf der Katgori-Ansicht funktionieren, und wenn ich dann 10 Produkte habe und überall wandert so in JavaScript-Monster, das wird mit der Zeit unkontrollierbar. Weil da auch immer wieder neu

Funktionalität n rang flanscht w rd n und wir das wartbar programmi r n ab r di Funktio-
nalität nicht m hr 100% sich rst ll n könn n.

Es ist j tzt im Lauf d s Proj kts b stimmt dr i od r vi r Mal passi rt, dass d r Kund
g sagt hat, „Ab r das F atur hat doch früh r schon mal funktioni rt.“ Das ist wirklich das
größt Probl m: wir könn n nicht d n ganz n Shop nach j d r Entwicklung durcht st n. Und
da li gt di größt Schwachst ll mom ntan b i Mag nto und b i uns r m Proz ss, di wir
schn llstmöglich lös n müss n.

FS: J tzt sprachst du g rad in Zusamm nhang mit JavaScript nochmal von Unit T sts, das
funktioniert schon?

GP: Ja. Das hab n wir noch nicht impl m nti rt ab r s funktioni rt, man kann auch JavaS-
cript Unit-t st n.

FS: Klar, das kann man. In Mag nto ist das JavaScript all rdings v rstr ut, mit s hr vi l n
Abhängigk it n, s ist nicht s hr modular. Od r ist das in d m Fall in kompl tt ig n s?

GP: Ja, das ist twas ig n s mit JQu ry drüb r g l gt, da ist nichts m hr aus Mag nto und
d sw g n könn n wir das probl mlos t st n.

FS: Di H rang h nsw is würd st du auch mpf hl n?

GP: Uns r Front ndl r s h n das als b st Möglichk it. Ich bin s lb r k in Front ndl r ab r
ich glaub d n n. Mag nto nutzt ja auch Prototyp im Standard, das ist glaub ich nicht
unb dingt di rst Wahl von JavaScript Entwickl rn.

FS: Wir hab n uns noch nicht s hr mit d r Impl m nti rung an sich b schäftigt, was gibt s
da für typisch Probl m (Auf das Fram work b zog n, nicht auf d n Proz ss)?

GP: Es gibt s hr s hr vi l Möglichk it n, in Sach zu impl m nti r n, in Probl m zu
lös n; und di H rausford rung ist imm r, in schön Lösung zu find n. Es gibt halt vi l
unschön Impl m nti rung n, zum B ispi l Cod im T mplat zu schr ib n.

Di groß H rausford rung ist das Fram work und das dahint r li g nd MVC-Patt rn, wirk-
lich zu nutz n, und di inz ln n Kompon nt n und Funktionalität n auf di richtig n B r ich
aufzut il n. Das ist d r Anspruch ab r gl ichz itig auch nicht so infach, w nn man s nicht
g wohnt ist.

Es ist halt unh imlich infach, im T mplat ob n Busin ss Logic auszuführ n und di Rück-
gab n davon auszug b n, wi s früh r b i PHP 4 halt war. Und das ist di H rausford rung,
di Programmbaut il di man schr ibt auf di inz ln n Kompon nt n zu v rt il n, di ja
in b stimmt Aufgab und V rantwortung hab n.

FS: Ist di s Tr nnung zwisch n T mplat und Programm nicht stark g nug g g b n od r
infach nicht forci rt?

GP: Es wird w nn man sich d n Cor anguckt, vorg g b n ab r nicht forci rt. Es gibt Fram -
works, di funktioni r n nicht mit Plain PHP im T mplat . In Mag nto kann man b n Plain
PHP schr ib n, w nn man will, das h ift, man kann all s mach n. In and r n Fram works
gibt s ig n T mplat Engin s, wi Fluid in Flow3. Da kann man k in PHP drin schr ib n,
da gibt s in vorg g b n Syntax di d r Entwickl r aufruf n kann und da kommt man gar
nicht in V rsuchung, PHP drin zu schr ib n. Da ist irg ndwo Mag nto nicht strikt g nug ab r

ich ziehe es vor, Plain PHP im Template zu schreiben. Ich würde es nicht begrüßen, da noch in Template-Engin vorzustellen.

FS: Jetzt erlaubt Symfony zum Beispiel Plain PHP Templates, ohne dass man von dort auf alles zugreifen kann, da alle Abhängigkeiten von Anfang an sehr klar festgelegt werden, wohingegen man in Magento von jeder Stelle aus über `Magento::pp()` an alle Bestandteile heran kommt. Ist das vielleicht das Problem, also dass es zu vielen Abkürzungen gibt?

GP: Ich glaube, in dem Moment wo ich PHP zur Verfügung habe, kommt man immer irgendwohin, ich habe auch schon `require 'Magento.php'` im Template geschrieben... In dem Moment, wo ich die Freiheit habe, gibt es immer Leute, die die Freiheit nutzen, weil sie es nicht besser wissen. Und das ist die größte Herausforderung dabei, dass ich mich als Entwickler an die Vorgaben des Frameworks halten muss. Sie werden nicht erzwungen, wie bei Flow3, was ich persönlich aber als Risiko-Vorteil sehe. Ich muss mich dran halten, klar, aber ich habe alle Freiheit, wenn ich sie mal braucht. Das ist für mich die größte Unterscheidung zwischen Zend Framework und Flow3, einmal die strikte Framework wo ich nur noch Klassen schreiben und Objekte, die miteinander agieren und einmal Zend Framework, aus dem ich einfach Komponenten nutzen kann, aber nicht in irgendwas rein drängen werden, in dem ich mich gar nicht aufhalten möchte.

FS: Flow3 bringt also mit der eigenen Template-Syntax noch in Abstraktion hinein, ansonsten ist es bei den PHP-Frameworks aber so, dass man immer Abkürzungen finden kann, um irgendwohin sauber zu arbeiten.

GP: Ich sehe das nicht als Nachteil, ich sehe es in der Verantwortung des Entwicklers. Ich sehe es nicht als Aufgabe des Frameworks, den Entwickler zu zurechtweisen, sondern der Entwickler soll sich in der Verantwortung bewusst sein, qualitativ hochwertige Software zu schreiben. Das heißt die Rollenvertauscht, wenn das Framework versucht, den Entwickler dazu zu zurechtweisen, sauberen Code zu schreiben. Wenn der Entwickler diesen Anspruch gar nicht hat, dann wird das auch im Framework nicht schaffen.

FS: Nun ist es gerade bei PHP herumkehrt, dass es in den nicht nur nicht dazu zurecht, sauberen Code zu schreiben, sondern eher dazu verleitet, auch mal unsauberen Code zu schreiben. Das ist bei Plain PHP oder älteren PHP-Anwendungen wahrscheinlich stärker der Fall als bei Magento aber merkt man das noch in der Entwicklung? Oder nur indirekt, wenn man sich mindertig Erweiterung von unfernen Entwicklern ansieht?

GP: Bei uns ist es so, dass der Leuten ist mit ihrem Plain PHP Wissen kommen und der Code total krautig ist. Man muss, wenn man in den neuen Entwickler bewerten muss, auch über das Potential sehen, sauberen Code zu schreiben als das was er momentan schreibt. Er muss hier den Anspruch haben, strukturiert zu arbeiten als direkt von Anfang an strukturiert zu arbeiten, weil wie du sagst, PHP es so einfach macht, unsauber zu schreiben. Aber wirst du schon sehr viel Engieren, den Programmieren saubere Arbeitswiese beibringen, und dann ist es auch möglich. Aber es ist halt immer so, dass man sich selbst dazu bringen muss, man muss selbst diesen Anspruch haben.

Ich komme selbst her aus der C# Ecke, habe früher mit .NET programmiert, deshalb habe ich mich auch in Magento oder Zend Framework wohl gefühlt als in irgendwelchen Typo3 oder alten PHP 4 Anwendungen. Klar ist es erstmal merkwürdig, dass die Variablenkonventionen Typ haben, daran muss man sich als Programmierer erstmal gewöhnen, wenn man vorher mit Java oder C# gearbeitet hat, also daran dass so in Variablen Typen springt oder springen

kann wenn man nicht aufpasst. Ich würde mit PHP wahrscheinlich auch kein Raketensoftware programmieren. Es ist genauso wie du sagst, dass PHP in mir nicht schwer macht, krautig zu programmieren.

FS: Jetzt habt ihr also die typische Situation, dass ihr neu Programmieren lernt, die ihr aus dem PHP-Eck kommt und die quasi in bisschen rühen müsst oder anlernen, sauber zu arbeiten. Gibt das alles über den Review-Prozess oder anfängliches Training oder wie läuft das?

GP: Am Anfang muss man sich noch persönlich sorgen, also wirklich hier so Mentor-mäßig dann sitzen und zeigen, wie gut der Code aussieht. Sonst rüht man zu oft in die Sackgasse, wenn man das nur über den asynchronen Review-Prozess macht. Der macht später Sinn, wenn man es weiß. Level erreicht ist, aber am Anfang muss man erst mal die Entwickler dazu bringen, zum Beispiel Magento-Code zu lesen, zu schauen, wie die Controller-Entwickler das machen, und auch da muss man darauf hinweisen, dass es teilweise nicht so schön programmiert ist.

Und mit schließlichem Beispiel kann man auch vilerleichen. Wir haben ja schon einige Projekte und wenn da in Entwicklung sieht, so macht man es nicht und auch sieht warum... wir haben Beispiel, zum Beispiel in dem Template wurde oben in SOAP-Client erzeugt, da wird dann die API von Magento angesprochen – man ist gerade selbst in Magento! – dann werden alle Customergesamtheit, und dann wird in foreach über alle Customergesamtheit, die man via API aus dem Shop gesammelt hat, indem man gerade ist, und dann für alle Customergesamtheit das Kennwort verglichen (wir sind beim Login). Das ist halt spannend, so etwas bleibt dem Entwickler dann auch im Kopf, und er sagt, so will ich nicht werden.

FS: Ihr habt also in Arsenal an Negativbeispielen und könnt daran auch zeigen, was die Auswirkungen sind?

GP: Ganz genau. Oft ist es ja so, dass gefragt wird, „Warum soll ich denn nicht im Template Business Logic schreiben?“, und da sieht man mal, was passiert, wenn man das schließlich lässt. Ich glaube, die Negativbeispiele sind ganz gut.

FS: Du sagst gerade „Mentor-mäßig dann sitzen“, nutzt ihr auch Pair Programming als Technik?

GP: Ja. Gerade beim agilen Scrum fördern wir das, sagen „Findet raus wie ihr am besten arbeiten“, und gerade wenn es zwei Entwickler gibt, die vom Skill her annähernd auf dem gleichen Level sind (das muss aber nicht mal, man kann da wie gesagt auch viel Mentor-mäßig machen), dass man sich zusammensetzt und gemeinsam Code schreibt. So zieht man sich auch gemeinsam auf in Level hoch. Das macht aber in meinem gewissen Status und Skill-Level auf jeden Fall Sinn und das machen wir auch.

FS: Und das funktioniert gut?

GP: Das funktioniert gut bei uns.

FS: Nachdem wir uns vorhin mit PHP beschäftigt haben: Du hast den Vergleich zu C#, also auch zu Desktop-Anwendung oder Client-Server-Anwendung. Gibt es in den beiden unterschiedlichen Entwicklung von webbasierten Applikationen im Vergleich dazu? Unabhängig von der Sprache?

GP: Eigentlich gibt es da auf dem Prozessbereich in unterschiedlichen. Wir könnten mit dem gleichen Methodengestaltung in Desktop-Anwendung bauen.

FS: Und dass es in Windows-Entwicklung ist stellt euch nicht vor zusätzlich Herausforderungen?

GP: Dadurch dass wir das was man von Desktop-Anwendungen hat, also Entwicklungsumgebung, Debugger usw. hier auch aufbaut haben, ist es genau das gleiche wie wenn ich jetzt mit dem Microsoft Visual Studio in Desktop-Anwendung baue. Es hat uns vorhin paar Herausforderungen gestellt, ja. Es ist schon so, dass wir uns erst mal dieses Environment aufbauen mussten, dass wir zum Beispiel immer das Client-System, was auch der Kunde hat immer auch bei uns haben. Also wenn er in Remote nutzt oder Debian, haben wir das auch bei uns integriert, sodass wir auch auf dem System auf dem die Software nachher läuft, entwickeln. Das ist schon relevant.

Dadurch dass jeder Entwickler auf seiner virtuellen Maschine bei uns auf dem Server arbeitet, hat jeder Entwickler immer bei jedem Projekt das Live-System annähernd integriert. Wenn das auf seinem Windows-Server läuft, läuft da halt irgendwo Windows Server (was wir nicht haben, zum Glück). Einfach um das auch lokal unter sich in der Konfiguration auszuschießen. Wenn ein Kunde bzw. in Admin vom Kunden sagt, maximal Skriptlaufzeit ist 60 Sekunden oder 30 Sekunden, dann kann man gucken: „Bei uns auf dem Dev läuft es, auf dem Live läuft es nicht, OK, und der Skriptlaufzeit“, und so haben wir das immer 1:1 integriert. Das sind schon irgendwo Beschränkungen, die wir uns erst mal aufbauen mussten, um ideal Bedingungen für die Entwicklung zu schaffen.

FS: Es ist also schon ein Problem, was ihr aber offensichtlich gut gelöst habt: Serverkonfiguration, Auswirkung von lokalen Änderungen, und so weiter.

GP: Klar, das ist immer noch nicht optimal. Wenn der Admin etwas ändert, bekommt wir das nicht immer zwangsläufig mit. Da gibt es nicht die perfekte Kommunikation oder man tauscht nicht immer perfekte Konfigurations-Skript aus. Wenn der Host etwas ändert, bekommt wir das teilweise auch nicht mit.

FS: Aber ihr versucht schon, auch mit dem Administrator jeweils abzustimmen, wenn denn in der zur Verfügung steht?

GP: Ja. In dem Fall ist ein Fall hat man ja schon Zugriff auf den Server, zumindest ist es so, und kann sich die Konfiguration dann auch ziehen, in gewissen Abständen.

FS: Ihr seid immer noch in der relativ klaren Unternehmung, gibt es Beschränkungen, die sich dadurch ergeben? Ich denke mal in Hinblick auf Scrum eher nicht, das ist genau darauf ausgelegt ist?

GP: Ja, also in Scrum Team sollte zwischen 5 und 7 Leute stark sein und ich glaube auch bei großen Unternehmungen ist so in Scrum Team nicht so viel größer oder man hat Sub-Scrum Teams. Ich glaube schon, dass es in Beschränkung ist, wenn man so klein ist, weil die Kommunikation einfacher ist. Wenn ich mir vorstelle, man sitzt hier mit vielleicht 200 Entwicklern, da gibt es in gewissen Kommunikations-Overhead einfach dadurch dass man so viel koordinieren muss. Ich glaube schon, dass es einfacher ist, dadurch dass wir noch nicht so groß sind und das alles noch überschaubar ist.

FS: Scrum macht ihr nun aber auch erst seit anderthalb Jahren, ihr habt also anfangs auch mit klassischen Modellen wie Wasserfall gearbeitet. Was waren da die Probleme und was hat sich mit dem Wechsel verbunden?

GP: Beim Wasserfall war das Change Request Management des Kunden für uns regelmäßig in der Herausforderung. Dass man am Anfang Spezifikationen hatte, die sich im Laufe des

Projekts geändert haben und die Kommunikation dann mit dem Kunden immer unlustiger wurde. Dies ist klassisch Beispiel: Der Kunde sagt, „Ich möchte einen farbigen Elefant, programmiert mir einen farbigen Elefant.“ Wir programmierten zwar in der Woche, programmierten ihn rot. Sagt der Kunde am Ende, „Wieso ist der rot, ich wollte doch blau. Hab ich doch gesagt.“ – „Nein, du hast gesagt, er soll farbig sein.“ Solche Gespräche haben wir dann immer öfter geführt und der Kunde sah dann überhaupt nicht ein, warum er jetzt von rot auf blau nochmal in der Tagbuchzahl soll. Wir denken halt auch, dass Veränderung im Lauf des Projekts zur Softwareentwicklung dazu gehört. Es ist eben kein Ingenieurwissenschaft im dem Sinne dass ich in Brückenbau und weiß ganz genau, die muss soundso viel Tonnenaushalten, deshalb müssen wir sie jetzt soundso planen. Wenn das Fundament der Brücke einmal steht, kann ich es nicht mehr ändern, bei Software kann ich im Prinzip das Fundament zu jeder Zeit zu jedem Moment im Projekt austauschen. Und deswegen war das bei uns nicht mehr zeitgemäß, weil die Probleme, die bei uns mit dem Wasserfall entstanden sind, immer größer wurden. Wir sahen da nicht wirklich in Zukunft drin, auf dieses Modell zu setzen.

Ich glaube schon, dass man gut Software mit Wasserfall bauen kann, wenn die Spezifikation wirklich fest ist und die Kommunikation mit dem Kunden funktioniert, aber auch da entwickelt sich dann so in Wasserfall in dem Moment wo Veränderung eintrifft, und wird auch wieder irgendwo agil. Dann verändert sich auch die Spezifikation und das macht Scrum einfach nur transparent.

Wir wollten mehr Transparenz bei uns, für uns und für unsere Kunden. Der Kunde sollte mehr im Mittelpunkt der Entwicklung stehen, nicht ganz am Anfang wo der Spezifikation eintrifft und ganz am Ende, wo er dann merkt dass die Spezifikation nicht erfüllt ist, sondern wir wollten den Kunden im Projekt mitführen und in regelmäßigen Abständen zeigen was gemacht ist, ihm die Möglichkeit geben darauf frühzeitig zu reagieren. Damit wir eben nicht auf in mehrmaligen Fundamentaufbau, sondern frühzeitig sagen können, „Guck mal, das ist das Fundament, ist das die richtige Richtung? Wir haben das jetzt blau gemacht, soll das blau?“ Was da jetzt alles dran hängt mit dem Vortragen usw.: OK, außen vor, das ist in anderer Sprache. Aber vom Entwicklungsprozess hat es sich deutlich verbessert, dadurch dass wir den Kunden in dem Mittelpunkt nehmen.

FS: jetzt sagst du, theoretisch könnte man auch so gut Software entwickeln, spielt da der Kontext von Magento / Web-Frameworks / E-Commerce eine Rolle?

GP: Ich glaube, man kann dann gut Projekt mit Wasserfall machen, wenn der Kunde wirklich weiß, was er will. Und Magento ist so unübersichtlich für den Kunden – die wenigsten Kunden wissen, weil Faktoren sie quasi schon haben. Es ist ja im Grunde in Standard-Software, nichts was von Grund auf neu gebaut wird. Er hat ja schon ganz ganz viel Möglichkeiten, die die Community oder Enterprise Edition bringen, das heißt er wird mehr in die Meinung nach zwangsläufig früher oder später zu dem Punkt kommen, an dem er sieht, „Ach, da ist ja noch etwas anders, das möchte ich ja auch noch können.“ Ich glaube Wasserfall funktioniert nicht mit Magento. Außen der Kunde kennt Magento sehr sehr gut und hat in sehr sehr gut Spezifikation ganz am Anfang, weil Faktoren sie zusätzlich möchte.

FS: Man könnte also sagen, es kommen üblicherweise Veränderungen im Lauf des Projekts weil man einfach nicht von Anfang an Überblick haben kann, was für Möglichkeiten es gibt, weil die Basis-Anwendung auch ist?

GP: Ja. Also dieses Standard-Faktor-Set ist in den allerwenigsten Fällen allen bekannt, die an dem Projekt beteiligt sind, dem Kunden am wenigsten.

FS: Ergibt sich daraus abgesehen von vielen Änderungswünschen noch andere Probleme?

GP: Das Problem ist ganz häufig, bei uns zumindest, dass der Kunde, der das Projekt beauftragt, erst mal nur der Marketing-Liter ist. Er nimmt die Leute, die damit wirklich arbeiten, zu spät mit, die Anwender, also die Reaktor oder diejenigen die die Produkte pflegen. Die werden zu spät in das Projekt mit aufgenommen, und dann fällt den Anwendern erst auf, „Wir möchten doch eigentlich noch das können und eigentlich haben wir uns noch das vorgestellt“... ich glaube das ist eher in Abstimmungsproblem aufgewissenen Situationen.

FS: Ist es vielleicht auch ein Problem für Entwickler, gerade nur?

GP: Teilweise wahrscheinlich schon. Du meinst, dass auch in Entwickler das nicht alles überblicken kann? Ja, teilweise definitiv. Auf Mallorca war das erstmal das Magento Developer Paradise und da waren auch Core-Entwickler und aus dem Publikum kamen Fragen zu PayPal oder zum Katalog-Index. Und er hat gesagt, „Nur ich mache nur CMS.“ Also selbst im Kernel-Team gibt es Leute, die die Software bauen, die sich aber nur mit Komponenten der Software auseinandersetzen, weil es so komplex ist; man kann nicht alles von Magentokennen. Und es ist definitiv so, dass auch in Entwickler sich erstmal in den Überblick verschaffen muss und erst mal in die Thema sich einarbeiten muss wenn es um Anforderungen geht. Definitiv.

FS: Wie läuft denn das zum Beispiel mit der Einarbeitung in die neuen Themen? Angenommen ich bekomme in User Story, wo ich weiß, es ist in Teil von Magento mit dem ich mich noch nicht befasst habe und weiß nicht, wie ich da vorgehen werde. Ich muss aber zunächst mal in die Schätzung abgeben.

GP: Ja. Die Schätzung ist in dem Moment wo ich mich mit der Komponente nicht so gut auskenne wahrscheinlich noch größer als wenn ich mich damit auskenne. Wenn man Scrum macht und das nur abstrakt in Story Points angeben muss, ist es nicht so das Problem. Wenn man dem Kunden aber in Ansagen machen muss, dauert zwei Tage oder dauert drei Tage, dann müssen wir schon in gewisser Weise in Vorlesung gehen indem wir die Story schätzen und uns dabei ein bisschen in die Komponente einarbeiten, um in den besten Überblick zu haben. Auch auf die Gefahr hin, dass der Kunde das im Endeffekt doch nicht beauftragt. Das ist halt bei so einer komplexen Software anders nicht machbar. Bevor wir dann zu groß schätzen und den Auftrag gefährden, gehen wir lieber die Risiken ein, uns ein bisschen in die Komponente einzuarbeiten um in den Überblick zu haben. Aber anders geht es nicht, weil das System einfach zu groß geworden ist.

FS: Also es kommt schon noch regelmäßig vor oder wird das mit der Zeit weniger?

GP: Mit der Zeit wird es weniger, aber es passiert gerade wenn man neu Entwickler hat, die sich noch nicht gut damit auskennen und dann etwas schätzen sollen, auch die Erfahrung des Schätzens machen sollen. Denn es sollte unser Verantwortung nach immer der Schätzer, der es macht. Und wenn der, der es vielleicht besser könnte, nicht kann – man will ja schon das Level des Teams gleichmäßig hochziehen – dann muss er sich einarbeiten.

FS: Zusammenfasst: Wo siehst du am höchsten Verbesserungsbereich und -potential im gesamten Entwicklungsprozess?

GP: Einmal sehe ich Verbesserungspotential bei der Schulung des Kunden, in Scrum zum Beispiel. Bei uns ist es oft so, dass der Product Owner, der ja eine wichtige Rolle im Scrum Prozess einnimmt, auf Kundensite ist. Und die neuen Product Owner, der vtl. noch gar kein Scrum gemacht hat, den müssen wir erstmal schulen, in Hinblick auf Scrum aber damit auch in

Hinblick auf den Entwicklungsprozess, damit ihm überhaupt bewusst ist welche Rolle, welche Verantwortung er hat. Das ist in dem Punkt, den man verabschieden muss, an dem man auch dran bleiben muss, denn es ist bei jedem Projekt auf dem in der Herausforderung, weil man den Kunden ja einbezogen möchte, in den Prozess. Man möchte nicht nur Spezifikation vornehmen, Produkt hinterlassen, sondern der Kunde soll ja die ganze Zeit mitschwimmen. Das bezieht sich auch darauf, dass die Anforderungen in Form von User Stories genau werden, aber nicht mehr nur „Ich will in den Checkout-Button“, sondern was soll passieren?

Dann ist das größte Thema eigentlich, das automatisiert ist. Das ist glaube ich, wenn man agil entwickelt, sehr sehr wichtig. Und es ist so momentan mit Magento nicht möglich und in unserem Prozess noch nicht integriert.

FS: Gerade in Hinblick auf Testen, würdest du dir zusätzlich Werkzeug wünschen, wo die vorhanden sind nicht ausreichen?

GP: Wir haben noch nicht so viel Erfahrung mit dem Jenkins Continuous Integration (CI) Server gemacht, es gibt da auch ein Tool von Atlassian. Das Problem bei diesem Modul ist, es kostet immer Geld und die Preispolitik von Atlassian ist dazu mlich geschickt. Die füllt rein in den an, mit 10 Usern kostet es 10 Dollar im Jahr und ab 11 dann 1000 Dollar und man ist ganz schnell über die Grenze hinaus. Und es gibt da in den CI Server namens Bamboo, der ist sehr gut integriert mit Jira und den Tools die wir sonst so nutzen. Ich wünsche mir ein stärkeres Integration aller unserer Tools, weil die Tools eigentlich nur Mittel zum Zweck sein sollten. Eigentlich möchte ich den Prozess forcieren, versuchen Scrum zu machen, agil zu entwickeln. Und die Tools sind halt da aber wir machen das nicht wegen den Tools, sondern wir machen das, um den Prozess zu machen, und die Tools müssen uns dabei unterstützen. Und in dem Moment in dem wir so viel aufgesplittet haben, in Jenkins, was nicht wirklich integriert ist mit Jira... da gibt es Verabschiedungspotential was die Tools angeht.

In Hinblick auf Testen: ich glaube, es gibt schon ein Mapping, was man machen kann, nur wir nutzen es noch nicht, das ist halt das größte Problem

FS: Für die Integration der Tools gibt es also theoretisch Möglichkeiten, das ist aber ein Preisproblem?

GP: Ja. Bis vor kurzem war es noch ein technisches Problem, weil die nicht mit Git klar kamen. Jetzt wo sie aber selbst in Git Repository Tool haben, was wir auch einsetzen, ist es auch möglich, aus Git heraus zu bauen. Das heißt, jetzt ist es nur noch ein Preisproblem und ein Zeitproblem das aufzulösen und zu konfigurieren. Es besteht aber auch ein gewisses Unsicherheitsmoment, wenn man nicht weiß, deckt es jetzt alle Features ab, die wir brauchen.

FS: Auch ein Problem der Einarbeitung?

GP: Ich denke schon, dass man dadurch Zeit investieren muss, und man muss die Prozess darauf wieder abstimmen. Man hat halt den Entwicklungsprozess und da sind wir auch noch in der Findungsphase weil sich da zu viel immer ändert und es gibt nicht die in der perfekten Prozess. Das ist viel noch im Fluss, viel nicht auch weil wir es so agil machen, da verändert es sich viel.

FS: Ein anderer Feld was Werkzeug angeht ist das Deployment. Gibt es da noch Probleme?

GP: Gut, für das Deployment haben wir uns selber Skript geschrieben. Beim Deployment sehe ich auch noch viel Verabschiedungspotential, weil wir die Deployments noch nicht separat stellen. Da wünsche ich mir so ein Vorgehen wie in der Docker-Anwendungsentwicklung, dass

man so in Paket hat, was der User nachher einfach auf seinem System installiert, und so etwas haben wir noch nicht. Dass man also sagt, wir haben hier in Package und dieses Package wird getestet und wenn es funktioniert, wird es auf dem Server installiert. Das würde ich mir noch wünschen.

FS: Jetzt gibt es ja gerade in PHP einige Ansätze in die Richtung, wie Composer oder ganz aktuell das PHP-Maven Projekt, das versucht Maven in die PHP-Welt zu bringen. Siehst du da die Chance, dieses Deployment zu vereinfachen.

GP: Ich habe mich damit noch nicht beschäftigt, da müsst man sich erstmal inarbeiten um da in Aussage darüber treffen zu können.

FS: Auf jeden Fall ist das in Gebieten, auf dem noch viel zu tun ist...

GP: Definitiv. Es ist kein Datenhochschibben auf dem FTP Server aber auch noch nicht so kontrolliert wie man sich in Deployment vorstellt.

FS: Fällt dir noch etwas ein, was ich übergegangen habe, etwas was du noch zum Entwicklungsprozess sagen willst?

GP: Das war ja schon ziemlich viel. Ich glaube, wir haben alles wichtig mal angesprochen.

FS: Dann noch in Frage zum Abschluss in Bezug auf Magento: Wo könnte man beim Framework selbst ansatz nehmen, um die Arbeit damit noch zu erleichtern?

GP: Also wenn es von Haus aus Unit-test wäre, das wäre super. Es würde uns norm helfen, wenn wir dann nur noch die einzigen Extensions testen müssen, wäre es glaube ich ein sehr guter Schritt in die richtige Richtung. Dieses Test Automation Framework habe ich mir noch nicht angeschaut aber das geht zumindest in die richtige Richtung.

Von der Architektur her finde ich es schon gut gelöst bei Magento. Es bietet Möglichkeiten, in dem Coreinzugriff, ohne den Core zu überschreiben, mit dem Event-System, andere Möglichkeiten, den Core zu erweitern. In dem Sinne bin ich da schon mit zufrieden mit der Grundlage auf der man Shopsystem baut vs. komplett neu anfangen.

Es sind inzwischen Komponenten, die aus welchen Gründen auch immer nicht so flexibel sind wie man sie gerne hätte. Das PDF-Modul ließ sich zum Beispiel mit DomPdf besser lösen. Es ist aber auch in Einschränkung, dass so ein Framework den Anspruch hat, auf möglichst vielen Plattformen zu laufen, also von möglichst wenigen externen Bibliotheken abhängig zu sein, die der Host installiert haben müsste. Das sind Entscheidungen, die so in Architektur auch beeinflusst, da kann man den Magento-Engineering nur gar nicht so bösen sein.

Ich glaube schon, dass die Basis von Magento gut ist und erweiterbar ist. Also aus dem Stand kann ich nicht sagen, das und das ist schlecht gelöst.

FS: Konkret Fäll würde mir schon einfallen, zum Beispiel lässt sich die Template Nummer im Kunden-Modul nicht zu einem optionalen Feld machen ohne Core-Klassen zu überschreiben, da das einfach hart kodiert ist. Oder an gewissem Stellen bei der Preis-Berechnung wo man direkt, hier muss doch in Event hinabrennen, da sind Dinge einfach starr vorgegeben.

GP: An vielen Stellen fehlende Events, das stimmt. Jetzt wo du es sagst, immer wenn in E-Mail verschiebt wird wird auch kein Event gefeuert. Das würde ich mir auch wünschen, ich wollte zum Beispiel wenn in Rechnung per E-Mail verschiebt wird, auch das PDF verschieben,

das li ß sich nicht über in Event lösen. Und man ist schnell dabei, irgendwelche E-Mail-Klassen zu überschreiben, was nicht so schön ist. Da ist schließlich schade, dass wenn man Verbesserungsvorschläge an Magento schickt, die nicht schnell genug im Core landen. Denn irgendwo in Event zusätzlich zu werden ist glaube ich nicht so schlecht für den Core, macht nichts kaputt, würde aber vielleicht Aufwand sparen.

FS: Es fehlt also auch an Öffnung gegenüber den Entwicklern.

GP: Ja.

FS: Das war es dann, dank für das Interview!

A.2.3 Paraphrasiertes Interview mit Überschriften

Projektmanagement

Thema: Abrechnung

Frage: Was sind die Erfahrungen mit Scrum?

Wir machen Scrum seit gut anderthalb Jahren und bis jetzt sind die Erfahrungen größtenteils positiv, wenn sich der Kunde wirklich darauf einlässt. Es gibt in der ganzen Hypothese um Scrum, so dass es Kunden gibt, die davon viel gut gesagt haben und sagen möchten.

Bei uns ist Scrum aber auch verbunden mit der Abrechnung über Time & Materials, für Scrum die beste Vertragsart. Für den Kunden muss klar sein, dass es auch Geld kostet, wenn er Sprint für Sprint immer mehr fordert.

Das ist so ein Schwingen: wenn man das dem Kunden nicht zu Beginn des Projekts ausdrücklich sagt oder es ihm nicht so bewusst ist, ist der erstmalig blind von dem Vorteil von Scrum, den wir ihm ja auch so verkaufen: es gibt bessere Software. Es kostet aber am Ende oft dann auch Geld, das ist aber in der allgemeinen Problematik des Kunden/Lieferanten-Beziehung.

Wir haben Shop-Betrieb als Kunden, die selbst auch Entwickler angestellt haben. Mit dem ist es viel einfacher, Scrum zu machen, weil sie selbst wissen, was Entwicklung bedeutet, was für ein Aufwand dahintersteckt, die Probleme, die das Marketing in der gibt, zu lösen und weil die internen Entwickler noch dazu meist sehr unstrukturiert arbeiten. Rein Shop-Betrieb sieht dann nur die Kosten.

Thema: Arbeiten ohne Prozess

(Fortsetzung)

Wir haben einen Kunden mit großem Webshop, der hat fünf Entwickler und wenn man die Entwickler fragt, wie sie arbeiten, was für Prozesse sie haben, kommt die Antwort „Wir, Prozesse?“ – es gibt da wirklich keinen Prozess, man schreibt sich die Sachen im Raum zu: Das Marketing schreibt „Wir brauchen einen Button in grün“, dann macht irgendjemand in der Entwicklung, der gerade frei ist, den Button in grün, egal wie lang er braucht. Und wenn er fertig ist, wird es auf Live geschoben, ohne irgendjemand in der Deployment-Prozesse, ohne Testing, ohne Testplan. Es wird in Updates gemacht, teilweise auf dem Live-Server irgendwas gefixt, dann müssen sie gucken, dass sie sich irgendwo noch ins SVN bekommen – wenn sie in sie gibt.

Also es ist teilweise da draußen echt spannend, wenn man sieht, wie auch große Shops im Hintergrund arbeiten. Aber es funktioniert, die Frage ist halt nur wie lang. Nicht ohne Grund kommen die ja auch auf uns zu und sagen „OK wir haben so viele Probleme, den Shop zu warten, so viele Probleme, neue Features anzukriegen, weil wir immer Angst haben müssen, dass wir an irgendjemandem Stellen was zerschließen, weil es so unübersichtlich geworden ist.“ Jedoch hat auf seine eigene Art in dem Spezialfall inprogrammiert und kein Traut sich mehr wirklich an den Code. Das ist so das größte Problem auch von alten Shop-Systemen wie z.B. xt:Commerce. Die sind dann glaube ich auch ganz froh wenn es so ein Ansatz wie bei Magento gibt, wo das ganz erstmal unübersichtlich aufgebaut ist, modular aufgebaut ist.

Thema: Deadlines vs. Team Velocity

Frage: Wie hat sich Scrum bei den internen Prozessen bemerkbar gemacht?

Es gibt nicht mehr die Deadlines sondern der Kunde hat nach jedem Sprint die Möglichkeit, zu sagen dass er mit dem Stand live gehen möchte. Wir müssen intern nur dafür sorgen, dass am Ende der Sprints die Sachen gut bestellt sind und alles funktioniert. Das hat bei uns schon dazu geführt, dass wir wesentlich ruhiger sind und in anderer Atmosphäre im Projekt herrscht. Es gibt nicht mehr den Fall dass es in Letzt von unfruchtlichen Issues gibt aber in der Woche ist Deadlines und da geht das Ding auf jeden Fall live.

Du hast gewisse Bausteine, die du in gewisser Zeit auch fertig haben musst, dann bei jedem Sprint committet sich das Team auf die Umsetzung von User Stories. Aber man geht mit weniger zeitlichem Druck heran, dafür mit mehr Qualitätsdruck, es soll am Ende der Sprints perfekt sein. Es liegt aber auch am Product Owner, der immer auf der Seite des Kunden sitzt, ob im Sprint erfolgreich wird.

Das Team weiß mit der Zeit, wie schnell es vorwärts kommt, man rechnet nach jedem Sprint die Team Velocity in Story Points aus. Jedes Story wird abstrakt in Story Points geschätzt und wenn das Team alle zwei Wochen (so lang geht im Sprint bei uns) 25 Story Points schafft committet sich das Team darauf. Schafft es einmal weniger, werden bei dem nächsten Sprint auch weniger Story Points in dem Sprint gezogen und so verläuft sich das mit der Zeit aus. Die Anzahl der User Stories im Sprint ist damit so gewählt, dass man sich nicht übernimmt und nur den Fokus auf die gerade wichtigen und schaffbaren Aufgaben hat. Das ist gut für den Kunden und auch gut für uns.

Thema: Team-Gedanke

(Fortsetzung)

Zum Fokussieren auf die aktuellen Aufgaben wird das Entwickler-Team bei Scrum zusätzlich innerhalb der Sprints von äußeren Einflüssen abgeschottet, es gibt keine kurzfristigen Anfragen und Anforderungen. Darüber macht sich der Product Owner Gedanken und sieht von dem Scrum Master abfangen, dass das Team während der Sprints vor diesen Störeinflüssen geschützt.

Frage: Und das Modell hilft, die Qualität zu erhöhen?

Ich glaube schon, weil man sich auf weniger Sachen konzentriert und weil der Team-Gedanke gefördert wird. Sobald ein Bug auftritt, fühlt sich jeder dafür verantwortlich, weil das ganze Team sich auf den Erfolg in den Sprints committet hat. Da gibt es z.B. keine Anschuldigungen zwischen Backend und Frontend Entwicklern sondern man versucht sich gegenseitig zu helfen.

Wenn es in einem Blocker gibt und das System steht, haben viele Scrum Teams in Sprints, da drückt man auf einen Button und es müssen sich alle aus dem Scrum Team für dieses Projekt inmitten vom Raum versammeln und sich beraten, wie man den Blocker am besten lösen kann. Irgend einer hat die Lösung und in anderer Reihenfolge noch in den Hinweis und so kommt man viel schneller zum Ziel als wenn der Blocker jemandem zugewiesen wird, der sich vielleicht auch gar nicht damit auskennt und erst mal rumfragen muss. Das ganze Team fühlt sich für das gute Produkt verantwortlich und das bedeutet für uns halt auch gute Qualität, ich glaube schon dass Scrum da hilft. Wir haben keine Sprints, weil wir relativ klein sind aber wir tun dann so als ob und treffen uns am Scrum Board.

Thema: Teamgröße

Frage: Ihr seid immer noch ein relativ kleines Unternehmen, gibt es Besonderheiten, die sich dadurch ergeben?

Ein Scrum Team sollte zwischen 5 und 7 Leuten stark sein, ich glaube das ist auch bei großen Unternehmen nicht viel größer, oder es gibt Sub-Scrum Teams. In einem kleinen Team ist die Kommunikation einfacher, bei 200 Entwicklern gibt es in den Kommunikations-Overhead dadurch dass so viel koordiniert werden muss. Bei uns ist das alles noch überschaubar und ich glaube, dass es dadurch einfacher ist.

Thema: Scrum vs. Wasserfallmodell; Transparenz

Wenn wir nicht Scrum machen, sondern im Wartungsvortrag sind, kommen die Anforderungen als Aufgaben rein, wenn wir Scrum machen, in Form von User Stories: Der Kunde beschreibt aus Sicht des Users, was passieren soll und wie es sauber technisch umgesetzt wird, sofern der Kunde nicht zusätzlich technische Vorgaben macht.

Sobald die Anforderung in den Sprint kommt, der Product Owner also entscheidet, dass sie im nächsten Sprint umgesetzt werden soll, diskutieren wir gemeinsam, wie sie gelöst wird und erstellen dazu entsprechende Aufgaben. Zum Beispiel sagt der Backendler, er muss in Extension mit Modell und Controller schreiben und macht sich die Aufgabe „Extension schreiben“. Das wird diskutiert aber absichtlich nicht als feste Spezifikation niedergeschrieben. Wir wollen in der Diskussion, global, zusammenentwickelte Spezifikation, kein niedergeschriebenes.

Frage: Aber es gibt niedergeschriebene Hinweise, was getestet werden muss, ggf. mit Sonderfällen?

Ja. Es gibt sogenannte Akzeptanzkriterien. Angenommen, die User Story lautet, der Kunde möchte sich in das Kundenkonto einloggen und für den Kundenbestellung aufgeben können. Dann steht darin nicht, wie wir das technisch lösen, aber es gibt zum Beispiel das Akzeptanzkriterium, dass die Software nur mit festem IP-Adresse des Händlers genutzt werden kann, der Kunde möchte sich also im Gebäudebereich befinden. Solche technischen Akzeptanzkriterien müssen wir auch als Vorgaben haben und entsprechend testen.

Frage: An welchem Punkt werden die aufgestellt?

Zu Beginn, sie sind Teil der User Story. Entweder schreibt der Kunde sie von sich aus auf oder sie ergibt sich im Gespräch. Wenn der Kunde dem Team mitteilt, er möchte nächste Woche die User Story umgesetzt haben, fragt das Team, welche Einschränkung es gibt oder schlägt so etwas wie die IP-Sperre vor und der Product Owner präzisiert die User Story daraufhin.

Scrum fördert die Kommunikation, auch untereinander. Man spricht über die Sache und schreibt sie auch irgendwo nieder, das ist aber nicht formalisiert.

Thema: Aufwandschätzung

Frage: Gibt es bei der Aufwandschätzung typische Probleme?

Die gibt es generell, unabhängig vom verwendeten Tool bzw. Prozessmodell. Der übliche Ansatz ist in Expertenschätzung, man lässt also den Programmierer den Aufwand in Tagen oder Stunden schätzen. Die klassische Antwort lautet dann „zwisch“, ob das jetzt Tage sind, oder Stunden.

Das Schätzproblem geht Scrum ganz gut an, dort wird nicht die Dauer sondern die Größgeschätzt. Den Kunden interessiert für ein einzelnes Anforderung auch meistens gar nicht, wie lang es dauert sondern nur in Größordnung von klein/mittel/groß: dauert es eher zwei Stunden, zwei Tage oder 20 Tage? Scrum hat dafür die Story Points von 1, 2, 3, 5, 8, 13, die sind nicht linear durchnummeriert weil der Aufwand ungefähr zu schätzen ist je größer er ist. Das genügt meistens schon, so dass der Product Owner z.B. sagen kann, ein 13 für den Button ist es nicht wert, das wird nach hinten geschoben. Im Laufe des Projekts stellt sich dann auch heraus, was z.B. in achtzig Minuten bedutet, je nachdem wie das Team zusammengesetzt ist.

Frage: Wie funktioniert das in der Praxis? Kommt es trotzdem zu Fehleinschätzungen?

Nicht oft aber es kommt ab und an dazu dass man sich komplett überschätzt, in Anpassung als klein einschätzt aber das System sich in ihm doch nicht so einfach macht. Bei Magento im speziellen und nicht man oft, da muss auf jeden Fall in Eventberitstheorie um etwas schön zu lösen und obwohl das offensichtlich Sinn machen würde, gibt es gerade dort kein

Frage: Der Aufwand wird wahrscheinlich größer, wenn man das dann trotzdem ordentlich machen will. Die Frage ist, wie fängt man diese Fehleinschätzungen auf?

Zitlich ist das mit Scrum kein Problem. Du hast zum Beispiel die Anforderung, die in zwei Wochen umgesetzt werden soll und in davon dauert deutlich länger als geschätzt so dass wir nur die in Story geschafft haben. Das drückt die Anforderung in zwei Stories aus dem Sprint heraus, sie müssen im nächsten Sprint bearbeitet werden.

Das ist das Timeboxing-Prinzip von Scrum: alles was man macht, wird irgendwie „getimboxt“, in Sprint z.B. auf in oder zwei Wochen, das Daily Standup auf 15 Minuten. Also egal was passiert, nach 15 Minuten gehen alle weg. Das gilt auch im Sprint: Wenn wir uns vollkommen überschätzt haben und nur in Story geschafft wird, dann ist das halt so.

Abbruchtechnisch ist es hier in Problem, man muss dem Kunden sagen, dass es wahrscheinlich tiefer wird und da kommt das Vertragswerk ins Spiel, das man vorher gut festzurufen sollte. So was passiert nun mal in der Software-Entwicklung, das kann man nicht immer ausschließen.

Frage: Es wird also auch nicht mitten im Sprint gesagt, wir setzen die Story Points jetzt besser mal hoch?

Richtig. Um bei dem Beispiel zu bleiben: die drei Stories sind mit eins, drei und fünf geschätzt, sie sind also ein Story Points im Sprint. Als erstes soll die Eins umgesetzt werden, die überschlingt jetzt aber den ganzen Sprint. Dann haben wir nur in Story Point geschafft, was aber auch heißt dass im nächsten Sprint nur ein Story Point Velocity geschätzt wird. Oder sagen wir, wir schaffen die Eins und die Drei und die Fünf fällt raus. Dann werden in dem nächsten Sprint nur vier Story Points hinzunehmen. So nähert sich zu viel und zu wenig geschätzt nach und nach an und im Laufe des Projekts bekommt man heraus, was die realistische Velocity des Teams ist.

Frage: Die Eins bleibt auch im nächsten Schritt eine Eins?

Die Eins bleibt immer ein Eins. Wir haben sie mit eins geschätzt und machen sie fertig. Dass sie im Endeffekt größer war, solche Schätzfehler bündelt Scrum im Laufe des Projekts aus.

Entwurf

Thema: Architektur

Frage: Wenn man mit einem Task beginnt, wird man ja erst mal irgendwie einen Entwurf machen. Was gibt es da für Probleme, oder was ist da vielleicht spezifisch für Magento- und PHP-Projekte? Es gibt sehr sehr viel Möglichkeiten, in Sachen zu

implimentieren, in Problem zu lösen; und die Herausforderung ist immer, in schön Lösung zu finden. Es gibt viele unschöne Implementierungen, z.B. Code im Template zu schreiben.

Die große Herausforderung ist, das Framework und das dahinterliegende MVC-Pattern, richtig zu nutzen, die einzelnen Komponenten und Funktionalitäten auf die richtige Weise zu verteilen, die ja in sich selbst stimmig und Verantwortung haben. Das ist nicht einfach, wenn man es nicht gewohnt ist.

Es ist dagegen unheimlich einfach, im Template oben Business Logic auszuführen und die Rückgaben davon auszugeben, wie man es früher mit PHP 4 gemacht hat.

Wir versuchen immer erst die sauberste Möglichkeit zu finden, in gewisser Weise schränkt uns das Magento schon ein, weil wir da nicht alle Möglichkeiten haben, die man sich so vorstellt und weil Magento gewisse Patterns vorgibt, die man nutzen kann, um Probleme sauber zu lösen. Beim Flow3 Framework gibt es Aspekte im Sinne der Aspektorientierten Programmierung, da kann man viele flexible Anpassungen vornehmen ohne in das System eingreifen zu müssen. Aber auch das Event-System von Magento hilft in dem schon beim Entwurf von Lösung von Anforderungen.

Man weiß, an bestimmten Stellen kann man sich anhängen, um die Anforderungen in die Standardinstallation einzubringen, ohne viel selbst programmieren zu müssen. Die Sachen, die Magento liefert, haben in dem schon, in der ordentlichen Entwurf zu machen.

Frage: Indem man die Patterns nutzt, die Magento bereitstellt oder selber nutzt. Es gibt soweit ich weiß keine offiziellen Guidelines dazu, man muss also sehen wie es im Kern gehandhabt wird. Also Guidelines, wie man Probleme löst, auf Entwurfs-Ebene.

Das stimmt, ja. Also da hilft es viel, zu gucken, wie Magento selbst an Probleme herangeht, gerade beim Enterprise Packag, was ja auf dem Mage-Package aufbaut. Da sieht man häufig, wie die Enterprise-Entwickler mit diesen Anforderungen umgehen müssen: Enterprise-Faktors hinzubringen ohne den Magento-Kern anzufassen.

Frage: Habt ihr da irgendetwas herausgearbeitet oder kennt ihr alle den Kern und die Enterprise Edition einfach gut genug?

Haben wir nicht. Aber in einigen Problemlösungs-Sitzungen im Wiki anzuliegen wäre mal in gut Idee!

Thema: Entwurf mit UML

Frage: Bei einer Eigen-Entwicklung würde man den Entwurf mit Klassendiagrammen, Sequenzdiagrammen und so weiter notieren. Sind das Werkzeuge, die hier noch Sinn machen?

Um ehrlich zu sein wirst du die bei uns kaum sehen, in der Praxis braucht man das kaum und es bezahlt in dem auch kaum jemand. Bei komplexen Modulen macht es schon Sinn, um in initialen Überblick zu haben, weil es Aktuelles gibt.

Bei dem Flow3-Projekt, an dem wir Team-Überrichter gearbeitet haben, haben wir mal in Überblick von Klassendiagrammen im Wiki gehabt. Das Framework macht ähnlich wie das Zend Framework kaum Vorgaben und da musst du dir auch in dem Überblick verschaffen.

Abstraktion bei Anpassungen für Magenta und bei Modulen für Kunden, also wenn wir nicht in den Modulen schreiben, dann machen wir das kaum.

Frage: Es ist ja auch so, dass die meisten Akteure Teil des Frameworks sind, man sich in die Hot Spots einhängt und es zwischen eigenen Klassen eher wenig Interaktion gibt.

Das ist richtig.

Thema: Verständnis

Frage: Stellt es ein Verständnis-Problem dar, dass man diesen Überblick vielleicht gar nicht so hat, da viel im Framework passiert? Weiß man trotzdem, was da passiert?

Magenta hat in überschaubarer Anzahl an Klassen und Objekten, aber wenn man genug Erfahrung hat, verliert man nicht so schnell den Überblick. Es besteht aber auf jeden Fall die Gefahr, weil das Framework an sich (ich sehe Magenta auch eher als Framework) über die Jahre sehr komplex und mächtig geworden ist. Es stellt in der doch immer wieder vor Herausforderung, da der Überblick zu behalten.

Ich könnte mir z.B. nicht vorstellen – und ich glaube, viele Magenta-Entwickler müssen das noch so machen und machen das so – ohne Debugger auszukommen. Da passiert einfach zu viele Sachen, es wird zu viel mit Events gearbeitet, als dass man da den Überblick behält, was zu jedem Zeitpunkt was macht.

Ich hatte heute morgen noch den Fall, dass ich mich gefragt habe, warum der Index gebaut wurde. Ich habe in der Indexverwaltung gesehen, dass der Index vor in der Virtualisierung gebaut wurde, habe nachgeschaut, was zu der Zeit passiert ist. Dann habe ich gesehen, dass in der Bestimmungsgang geändert wurde. Und bei der Änderung der Bestimmung wird der Stock wieder ins Hoch gesetzt, und das hatte zur Folge, dass neu indiziert wurde. Ganz viele Sachen passieren überhaupt nicht linear, da muss man einfach mit dem Debugger gucken, was passiert.

Implementierung

Thema: Technische Schuld

Frage: Die einfachste Lösung ist oft nicht die sauberste. Was heißt das für die Implementierung? Man kann ja viele Sachen ganz dirty im Template machen und ist damit erstmal schnell fertig, schafft sich aber ein Problem, die dann irgendwann zurückkommen. Das ist in Problemen, was auch Scrum adressiert: die technische Schuld.

Also Scrum sagt, „Wenn ihr etwas löst, löst es sauber. Habt keine Angst vor Refactoring, schreibt den Code, den ihr jetzt braucht sauber, gut, gestutzt. Habt keine Angst davor, dass ihr ihn vielleicht im nächsten Sprint komplett über den Haufen werfen müsst um die dann beste Lösung zu finden.“

Wir hatten das Problem jetzt bei einem Projekt, das seit Januar dieses Jahres läuft: im zweiten oder dritten Sprint haben wir in Anpassung gemacht, die gerade so auf der Kippe zu dirty war. Und jetzt irgendwann in Sprint 23 merken wir doch, dass die Anpassung die

wir da g macht hab n, uns in n krass n Bug v rursacht hat. Di Z it di wir uns da g spart hab n, müss n wir j tzt dr i mal ins tz n um d n Bug wi d r saub r zu ntfri m ln.

Thema: Pair Programming

Frage: Nutzt ihr Pair Programming als Technik?

Ja. G rad b im agil n Scrum förd rn wir das, sag n „Find t raus wi ihr am b st n arb it t“, und g rad w nn s zw i Entwickl r gibt, di vom Skill h r annäh rnd auf d m gl ich n L v l sind (das muss ab r nicht mal, man kann da auch vi l M ntör-mäßig mach n), dass man sich zusamm ns tzt und g m insam Cod schr ibt. So zi ht man sich auch g m insam auf in L v l hoch. Das macht ab in m g wiss n Status und Skill-L v l auf j d n Fall Sinn und das mach n wir auch. Das funktioni rt gut b i uns.

Test

Thema: Unit Tests

Frage: Nutzt ihr Unit Tests? N in.

Frage: Warum nicht? Das ist in gut Frag . Wir woll n ig ntlich all s Unit-t st n und für in Ext nsion, di wir g rad s lbst schr ib n, schr ib n wir auch Unit T sts. Ab r s ist halt doch rh blich r Aufwand, d r l icht unt rschätzt wird, j d Kl inigk it, j d Unit zu t st n.

Wir hab n s b i in m Typo3-Proj kt g macht, da hat d r Kund s b zahlt, und da hab n wir uns auch s hr v rschätzt. Es ging glaub ich um 10 Tag Entwicklung, wir hab n 5 Tag für Unit T sts g schätzt, hab n im End ff kt initial für di Unit T sts 7 od r 8 Tag g braucht. Das war uns r rst s Proj kt wo wir das durchg zog n hab n, das sollt dann auch 100% s in, da hab n wir kompl tt t stg tri b n ntwick lt. Das in m Kund n zu v rkauf n ist unh imlich schw r.

In d m Fall war s in Ri s n-Kund und das li f üb r in groß Ag ntur, di g nug Budg t hatt und d n Vorschlag gut fand. Ab r das in m Mitt lständl r zu v rkauf n, ist glaub ich nicht so infach.

Frage: Hast du einen Vergleich, wie Unit Tests mit einem Anwendungsframework zusammen funktionieren, im Vergleich zu eigenen Anwendungen?

N in, da hab ich k in Erfahrungsw rt . Ich w iß, wi s b i Mag nto aussieht, da wir ja j tzt in Ext nsion ang fang n hab n, mit Unit T sts zu v rs h n. Das st llt uns noch vor in paar H rausford rung n, g rad was Controll r und so twas ang ht, in Abd ckung von 100% zu hab n, ist nicht so infach. Da arb it n wir noch dran. Normal Klass n sind ab r r lativ infach zu t st n.

Frage: Nutzt ihr dazu eins der Magento Unit Test Frameworks?

N in, PHPUnit ohn solch Erw it rung n.

Frage: Wie handhabt ihr dann die vielen Abhängigkeiten zu Magento, die man meistens noch hat?

Das war für uns klar in Problem, weil uns der Extension gar nicht so viel dirkt mit Magento zusammenarbeitete. Der Anteil von Objekten, die dirkt aus dem Framework kommen, ist relativ gering.

Das war die Extension, die Magento mit Typo3 verbindet und der Anteil, der da auf Magento-Site passiert, also mit dem Magento-Objekten wirden CMS Pages, ist vernachlässigbar.

Frage: Es gibt also Typen von Units, die einfach zu testen sind, weil sie sehr wenig Interaktion mit dem Framework haben und da würdest du es auch empfehlen?

Ja, und ich glaube da hat uns auch PHPStorm gut unterstützt, denn damit kann man die Tests gut initialisieren. Ich glaube deswegen sind wir auch um die Magento-spezifischen Unit Test Framework herum kommen. Wir initialisieren so bei jedem Test das ganze Framework, und haben Zugriff auf die zutestenden Objekte.

Thema: Funktionale Tests

Frage: Was für Werkzeuge nutzt ihr für funktionale Tests?

Das ist momentan alles manuell. Es wäre super-schön, automatisch Tests zu haben, gerade weil sie bei agiler Entwicklung eigentlich zum Pflichtprogramm gehört, aber das macht uns Magento nicht einfach. Das Framework ist eben nicht von Haus aus Unit-testet, im Gegensatz zu Oxid, z.B.

Unter funktionalen Tests würden auch Selenium Tests fallen, da sind wir auf dem Weg dahin. Wir nutzen sie noch nicht, für ein großes Projekt, was momentan in der Finalisierungsphase ist, werden sie kommen. Da streben wir an, dass die wichtigsten Komponenten des Magento-Shops funktional getestet werden. Dafür muss aber auch das passende Projekt dazu in und der Kunde die Notwendigkeit sehen.

Frage: Wie laufen die manuellen Tests denn ab? Wenn in Entwicklung ein Bug gefixt hat oder in der User-Story implementiert hat, geht das in die Qualitätssicherung. Ersetzt den Task am Scrum-Board auf „QA requsted“, dann geht in der Entwicklung für das Vier-Augen-Prinzip hin, und schickt ihn auf „in QA“, das heißt, getestet sie jetzt gerade. Wogetestet wird, da haben wir in der definierten Vorgabe. Er guckt sich den Code an, testet die Funktionalität wie von dem Entwickler programmiert hat beschrieben und von dem Entwickler den Bug oder das Feature erfasst hat, prüft ob die Anforderungen erfüllt sind. Wenn es in Feature ist, schreibt der Entwickler das implementiert hat auch in der Anleitung, was getestet werden soll, wie man auch die Eckfälle testen kann und der Tester prüft dann, ob das alles so funktioniert.

Frage: Testet ihr auch in diesem Qualitätssicherungs-Schritt, ob Seiteneffekte an unerwarteten Stellen auftreten oder wird das separat gemacht?

Da sprichst du in schwieriges Thema an. Es ist tatsächlich so, dass am Ende von diesen Umsetzungen im Prinzip alles nochmal getestet werden muss, und das ist momentan in sehr großer Herausforderung für uns, weil wir keine automatisierten Tests haben. Deswegen sind sie auch in dem nächsten Monat in den ganz großen Fokus darauf, dies einzuführen. Denn das ist das größte Problem bei uns, dass Seitenfehler, die durch Anpassungen passieren, für uns mit der Zeit unkontrollierbar werden.

Wir haben jetzt in den Fall, wo wir für in Produktkonfiguration in der JavaScript-Monsterbauern mussten. Es ging darum, alle Abhängigkeiten zwischen den 500 Einzelpro-

dukt n für d n Us r sichtbar zu mach n. Wir komm n nicht drumh rum, di s s JavaScript zu Unit-t st n und di S it mit S l nium zu t st n, di man dann nachh r in d r Kat gori -Ansicht si ht. Di s ganz n F atur s soll n nämlich auch auf d r Kat gori -Ansicht funktioni r n, und w nn ich da dann 10 v rschi d n Produkt hab und üb r all s wand rt so in JavaScript-Monst r, wird das mit d r Z it unkontrolli rbar.

Es w rd n imm r wi d r n u Funktionalität n rang flanscht und wir programmi r n das zwar wartbar ab r könn n di Funktionalität ohn T sts nicht m hr 100% sich rst ll n.

Es ist j tzt im Lauf d s Proj kts b stimmt dr i od r vi r Mal passi rt, dass d r Kund g sagt hat, „Ab r das F atur hat doch früh r schon mal funktioni rt.“ Das ist wirklich das größt Probl m: wir könn n nicht d n ganz n Shop nach j d r Entwicklung durcht st n. Und da li gt di größt Schwachst ll mom ntan b i Mag nto und b i uns r m Proz ss, di wir schn llstmöglich lös n müss n.

Thema: Code Review

Frage: Was gehört noch zum Qualitätssicherungs-Prozess?

Wir hab n noch in n Cod -R vi w-Proz ss, d r ab r asynchron läuft, also w nn d r d r s dann von d r Funktion h r g t st t hat, sagt, s funktioni rt, ist das Tick t „don “. Asynchron rz ugt d r, d r d n F hl r g fixt hat, od r das F atur impl m nti rt hat, in Cod R vi w und lädt dazu all Entwickl r in, di mit d m Proj kt was zu tun hab n, möglichst auch Entwickl r, di vom Skill h r üb r ihm sind. Er möcht ja auch w it r komm n und l rn n, da ist so in Cod R vi w Proz ss ganz hilfr ich, w il sich das T am so l v lt und hochzi ht. B i d m Cod R vi w wird üb rhaupt k in Funktion g t st t sond rn l diglich Cod -Qualität, Coding Guid lin s, Stil.

Das ultimativ Zi l ist ja, dass man sich vom T am h r so annäh rt, dass all gl ich Cod schr ib n. J d r kann d n Cod sofort l s n, d nn r hätt s auch s lbst s in könn n, d r s g schri b n hat.

Das R vi w ist b i uns in wichtig r Proz ss, da man dab i auch nochmal F hl r im Cod find t, di so vtl. nicht rsichtlich sind (z.b. di klassisch if-B dingung mit nur in m Gl ich-z ich n. Kann sup r funktioni r n, d r Ausdruck ist halt tru , und so twas fällt dann b im Cod R vi w rst auf).

Frage: Das Code Review wird tatsächlich für jeden Task einzeln mit dem ganzen Team durchgeführt?

Ja, g naug nomm n wird für j d n Commit in Cod R vi w ang l gt, bzw. b i Git j d s Mal w nn ins z ntral R pository g pusht wird. Das R vi w wird in d r Tracking Softwar auch dir kt mit d m zug hörig n Tick t v rbund n, g nau wi di commits.

Di L ut schau n sich j d r für sich d n Cod an, schr ib n Komm ntar und n hm n Än- d rung n vor. Di Komm ntar müss n nicht imm r ganz rnst s in, w rd n ab r schon als Komm ntar (g lb) od r F hl r (rot) klassifizi rt. So twas wi „Klass n-Komm ntar f hlt“ fällt b im Programmi r n und auch b im funktional n T st n nicht auf, d nn s funktioni rt ja all s. Das fällt rst auf, w nn man sich d n Cod anguckt.

Thema: Tests aus Kundensicht

Frage: Der Kunde muss die Notwendigkeit für automatisierte Tests sehen. Er hat also Einfluss darauf, je nachdem ob er sie will oder nicht? Der Gedanke dass ihr

mit Tests vielleicht auch effizienter entwickeln könnten spielt keine Rolle?

Es fällt bei diesem Projekt jetzt so langsam auf, weil das hier sehr komplex komponiert sind, und deshalb manchmal wir sind dem Kunden. Es ist halt trotzdem ein Problem, wenn der Kunde nicht zahlt, das dann auf die eigenen Kapazitäten hinweist. Wir würden, da gibt es dir vollkommen recht, viel effizienter und besser arbeiten aber so lang der Kunde das nicht versteht und sagt, „Egal, bevor du jetzt Tests schreibst, die ich nicht verstehen, zahl ich lieber fürs Bugfixing.“ – da sind wir dann doch Dinstag.

Das ist aus Sicht der Entwickler-Sicht überhaupt nicht nachvollziehbar. Der Kunde spart im Endeffekt, hat weniger Fehler, muss weniger fürs Bugfixing bezahlen, sein Entwickler haben auch mehr Spaß. Aber die Entscheidung liegt bei ihm und da steckt man manchmal nicht drin.

Ich glaube, in Zukunft wird man in den Shops ohne automatisierte Tests vielleicht gar nicht mehr verkaufen, man sagt: „Wenn ihr eine dirty Lösung wollt, sucht euch jemand anderen aber von uns erwartet ihr Qualität und die können wir euch nur liefern, wenn wir die Tests als Teil des Standard-Paketes ansuchen. Ihr wollt gut Software und wollt damit verkaufen, das geht einfach dazu.“ An dem Stand sind wir noch nicht aber ich glaube schon, dass wir dahin gehen müssen.

Nicht ohne Grund kosten die großen Shop-Systeme zehntausend an Lizenzgebühren. Ich glaube, die Branche muss weg von diesem Hinterrhof-Image, es wird langsam Zeit nach dem anfänglichen Hype um Magento in gewisser Professionalität hinzubringen. [...]

Integration

Thema: Wartung

(Fortsetzung)

Man hat ja als Magento-Entwickler immer noch genug Anrufe von anderen Agenturen, die Shops für 3000€ verkauft haben, die „irgendwie“ nicht funktionieren. Die kleinen Agenturen sind oft total überfordert von der Komplexität dieser Software.

Frage: Eure Projekte macht ihr aber von Anfang an selbst oder habt ihr auch solche Übernahmen von „missratenen“ Shops?

Nein, üblicherweise machen wir die Shops neu, so etwas sind nur ab und an kleine Sachen, wo Hilfe benötigt wird. Die macht man nicht immer gerne, weil es schon ein undankbarer Aufgabe ist, aus so einem verhassten Shop etwas sinnvolls zu machen. Wenn man da nicht unbedingt viel Aufwand investiert, bleibt immer irgendwo noch ein missratenes Stück und das fällt dann doch auf einen zurück, dann heißt es: „Ihr habt den Shop doch gefixt und das ist trotzdem noch ein Fehler.“

Man kann nicht alle Fehler finden, die da irgendwo programmiert wurden oder die vielleicht sogar schon im Code sind. Fehler gehören halt irgendwo doch zur Softwareentwicklung dazu, man wird kein 100% fehlerfrei Software herstellen außer vielleicht man schreibt Software für Flugzeug-Cockpits.

Thema: Fremd-Extensions

Frage: Noch mal zum Thema „unerwartete Seiteneffekte“: Die treten ja auch und gerade bei der Integration von verschiedenen Modulen auf, wo man nicht genau weiß, was im Hintergrund passiert und wo das noch Einfluss haben könnte. Erzähl mal etwas zur Integration von Fremd-Extensions.

Was die Qualität von Fremd-Extensions angeht, ist man immer wieder erstaunt, dass Leute überhaupt Geld dafür verlangen, was sie da verkaufen. Das sind teilweise Firmen, Agenturen, die selbst Shops implementieren und integrieren, sich aber an keine Coding Guidelines halten.

Da fragt man sich, ob es nicht doch sinnvoll gewesen wäre, das ganze selbst zu schreiben. Der Code funktioniert vielleicht, ist aber überhaupt nicht mehr wartbar und bringt somit letztlich endlich mehr Aufwand mit sich. Man hat ja die Verantwortung dafür, dass der Code auch in zwei Jahren oder nach einem Update noch funktioniert.

Die großen Herausforderung bei der Integration von Fremd-Extensions in Magento ist, sich anzuschauen wie sie aufgebaut sind und dabei Konflikte zu entdecken. Das Rewrite-System ist ja auch fehleranfällig, sobald zwei Extensions dieselbe Klasse überschreiben, haben wir ein Problem.

Frage: Du sprichst das Konfliktpotential des Rewrite-Systems an. Es ist auch bei richtig guten Extensions der Fall, dass Konflikte auftreten oder sie nicht zusammen funktionieren. Wie geht ihr da heran, diese Konflikte zu entdecken und vielleicht zu beheben?

Es gibt Workarounds, zumindest wenn die Rewrite-Systeme so aufgebaut sind, dass sie wenig verändern. Man kann dann die Klasse der in der Extension von der originalen trennen lassen. Der Tradeoff dabei ist, dass man sich die Upgrade-Fähigkeit kaputt macht und das ist halt unschön. Man muss irgendwo dokumentieren, schafft sich neue Abhängigkeiten.

Wenn Klassen komplett umgeschrieben wurden, so dass da gar nichts mehr ist wie vorher, dann nutzen wir die Extensions auch einfach nicht. Wir achten schon darauf, dass wir uns nicht allzu hohes Risiko ins Projekt holen. Wenn wir die Extension gekauft haben, geben wir sie zurück und lösen die Anforderung dann selbst auf saubere Art. Wir versuchen auch nicht, zu viele Workarounds zu machen.

Ich muss ehrlich sagen, wir nutzen nicht allzu viele Fremd-Extensions und versuchen immer alles zunächst auf unsere Art zu lösen wenn es nicht zu aufwändig ist. Bei krassen Anforderungen für die schon in gute Extension für 500 € oder 150 € gibt, schauen wir die an und probieren sie aus.

Das hatten wir zum Beispiel bei plus gemacht, in der Extension zur Anbindung an Prisma-Maschinen. Aber im Endeffekt hatten wir auch da so viel Aufwand hineinsteckt, die Extension aufzuräumen, da hätten wir sie auch von Anfang an selbst schreiben können.

Also wir haben wirklich schlechte Erfahrung mit Fremd-Extensions gemacht. Es ist ein großes Problem, dass die Qualität überhaupt nicht stimmt, die man sich da holt.

Thema: Infrastruktur; Deployment

(Fortsetzung)

Zum Thema Integration: Wir arbeiten mit Modman, weil es aus unserer Sicht ein großes Problem ist, dass die Daten in den Modulen über die Anwendung verstreut sind. Mit

Modman tr nn n wir d n K rn kompl tt von d n Erw it rung n und hab n si auch physisch an unt rschi dlich n Ort n. So hab n wir in d utlich b ss r Upgrad -Fähigk it für K rn und Modul . Di Modul könn n auch in in m xt rn n Syst m li g n, z.B. in Form von Git-Submodul n od r SVN-Ext rnals. Erst b im D ploy m nt w rd n K rn und Modul dann v rschmolz n.

Frage: Testet ihr die Module auch isoliert, also auf einem eigenen „sauberen“ System oder erst alles zusammengefügt?

Wir t st n all s dir kt im Proj kt zusamm ng fügt. Natürlich nicht auf d m Liv -S rv r od r Stag -S rv r, j d r Entwickl r hat s in Umg bung. Wir hab n in n groß n S rv r mit ganz vi l n virtu ll n Maschin n und j d r Entwickl r v rbind t sich auf s in VM, ntwick lt dort und kann auch dort Int grationst sts mach n. W nn das funktioni rt, g ht s w it r.

Frage: Wie sieht es mit dem Deployment aus, gibt es da noch Probleme?

Für das D ploy m nt hab n wir uns s lb r Skript g schri b n, da s h ich noch vi l V rb ss rungs-Pot ntial, w il wir di D ploy m nts noch nicht s parat t st n. Wünsch n würd ich mir da in Vorg h n wi b i d r D sktop-Anw ndungs ntwicklung, dass man in Pak t hat, was als solch s g t st t wird und d r Us r kann s infach auf s in m Syst m bzw. d m S rv r installi r n.

Frage: Jetzt gibt es ja gerade in PHP einige Ansätze in die Richtung, wie Composer oder ganz aktuell das PHP-Maven Projekt, das versucht Maven in die PHP-Welt zu bringen. Siehst du da die Chance, dieses Deployment zu vereinfachen.

Ich hab mich damit noch nicht b schäftigt, da müsst man sich rst mal inarb it n um da in Aussag drüb r tr ff n zu könn n.

Frage: Auf jeden Fall ist das ein Gebiet, auf dem noch viel zu tun ist...

D finitiv. Es ist k in Dat i n hochschib n auf d n FTP S rv r ab r auch noch nicht so kontrolli rt wi man sich in D ploy m nt vorst llt.

Frage: Gibt es sonst, unabhängig von der verwendeten Sprache, einen bedeutenden Unterschied bei der Entwicklung von webbasierten Applikationen im Vergleich zu Desktop-Anwendungen oder Client-Server-Anwendungen?

Auf d n Proz ss b zog n ig ntlich nicht, wir könn t mit d n gl ich n M thod n auch in D sktop-Anw ndung bau n. Di Entwicklungsumg bung mit D bugg r usw. hab n wir uns hi r auch so aufg baut dass s das gl ich ist, wi w nn man mit Visual Studio in D sktop-Anw ndung baut.

Ab r di s s Environm nt musst n wir uns rst inmal aufbau n, und das hat uns schon vor in paar H rausford rung n g st llt. Zum B ispi l, dass wir imm r das Syst m b i uns g spi g lt hab n, was auch d r Kund hat, auf d m di Softwar nachh r läuft. Das ist schon r l vant.

J d r Entwickl r hat b i j d m Proj kt das Liv -Syst m annäh rnd auf s in r VM g spi g lt, infach um auch kl inst Unt rschi d in d r Konfiguration auszuschli ß n.

Frage: Serverkonfigurationen und die Auswirkung von kleinsten Änderungen sind also schon ein Problem, was ihr aber gut gelöst habt?

Das ist immer noch nicht optimal. Wenn der Admin die Kunden etwas ändert, bekommen wir das nicht zwangsläufig immer mit. Es gibt also nicht immer die perfekte Kommunikation oder den perfekten Austausch von Konfigurations-Skripten. Auch Änderungen durch den Host bekommen wir vtl. nicht mit. In den allermeisten Fällen haben wir aber zumindest lückenlosen Zugriff auf den Server und können uns die Konfiguration dann in regelmäßigen Abständen ziehen.

Thema: Tool-Integration

Frage: Würdest du dir zusätzliche Werkzeuge wünschen, wo die vorhandenen nicht ausreichen?

Ich wünsche mir in stärkerer Integration aller Tools, weil die Tools eigentlich nur Mittel zum Zweck sein sollten. Das primäre Ziel ist, Scrum zu machen, agil zu entwickeln. Die Tools sind da, sollen uns dabei unterstützen, den Prozess zu forcieren, aber wir machen das nicht durch die Tools weg. In dem Moment, in dem wir so viel aufgesplittet haben, durch die Jenkins oder nicht wirklich mit Jira integriert ist, gibt es Verbesserungsbahnen.

Wir haben noch nicht so viel Erfahrung mit dem Jenkins Continuous Integration (CI) Server und es gibt auch in Alternativen namens Bamboo von Atlassian, der sehr gut mit Jira und unseren anderen Tools integriert ist. Bis vor kurzem kam Bamboo aber noch nicht mit Git klar und jetzt ist es ein Zielproblem, das aufzulösen, zu konfigurieren und zu evaluieren ob wirklich alle Features abgedeckt werden, die wir benötigen. Dazu kommt, dass die Preispolitik von Atlassian ziemlich geschickt ist: für 10 User kostet es 10 Dollar im Jahr, ab 11 dann 1000 Dollar. Über die Grenze ist man schnell hinaus.

Frage: Es ist auch ein Problem der Einarbeitung?

Ich denke schon, dass man dadurch ziemlich Zeit investieren muss, und man muss die Prozesse darauf wieder abstimmen. Man hat halt den Entwicklungsprozess und da sind wir auch noch in der Findungsphase, weil sich da zu viel immer ändert und es gibt nicht die selben perfekten Prozesse. Das ist viel noch im Fluss, weil nicht auch weil wir es so agil machen, da verändert sich viel.

Frage: In der Tracking Software werden Review und Commits mit zugehörigen Tickets verbunden, es gibt also schon Tools, die miteinander kommunizieren?

Die Tools von Atlassian sind bereits gut miteinander integriert: Jira für das Ticket Management, Stash für die Verknüpfung mit dem Git Repository (im Prinzip Github hinter der Firewall), Crucible für die Reviews.

Unsere Features entwickeln wir in Feature-Branches, die wenn sie fertig sind, in den Branch „develop“ gemergt werden. Die Commits werden beim Ticket verlinkt und umgekehrt, dazu muss man nur die Ticketnummer in die Commit-Nachricht schreiben.

Der Jenkins CI Server ist noch in der Test-Phase, der wird für die Systemtests genutzt, die immer wieder neu gebaut werden. Darüber könnten wir auch Selenium-Tests triggern oder Code Sniffer laufen lassen. Das ist aber in der Praxis noch nicht perfekt.

Sonstiges

Thema: Anwendungsframeworks

Frage: Du sagtest, man könne theoretisch auch mit dem Wasserfall-Modell gute Software entwickeln, spielt da der Kontext von Magento als E-Commerce Web-Framework eine Rolle?

Ich glaube, man kann dann gut Projekt mit Wasserfall machen, wenn der Kunde wirklich weiß was er will. Magento ist mit der Basis-Funktionalität schon so unübersichtlich für den Kunden, dass die wenigsten wissen, welche Features sie bereits in der Community oder Enterprise Edition haben. Früher oder später kommt er also zwangsläufig zu dem Punkt, wo er etwas neues entdeckt und sich neue Erweiterungen wünscht. Ich glaube, Wasserfall funktioniert nicht mit Magento, außer der Kunde kennt die Software sehr sehr gut und hat von Anfang an in sehr sehr guter Spezifikation über die zusätzlichen Features.

Frage: Man könnte also sagen, es kommen übermäßig viele Änderungswünsche im Laufe des Projekts, weil man nicht von Anfang an überblicken kann, was für Möglichkeiten es gibt und wie die Basis-Anwendung aussieht?

Ja. Das Standard Feature Set ist in den allerwenigsten Fällen allen bekannt, die am Projekt beteiligt sind, dem Kunden am wenigsten.

Frage: Ergeben sich daraus abgesehen von vielen Änderungswünschen noch andere Probleme?

Beispielsweise das Problem, dass der Kunde, der das Projekt beauftragt, häufig nur der Marketing-Liter ist. Die Anwender, die wirklich mit dem System arbeiten, werden zu spät in das Projekt mit aufgenommen, also die Reaktor oder diejenigen die die Produktpflege machen. Der Wunsch müsste eigentlich von Anfang an berücksichtigt werden, ich glaube das ist eher in Abstimmungsproblem aufgewissenes Set.

Frage: Ist mangelnder Überblick vielleicht ein auch ein Problem für Entwickler, gerade neuere?

Teilweise definitiv. Beim ersten Magento Developer's Paradise waren auch Core-Entwickler zu Gast. Da kam es aus dem Publikum Fragen zu PayPal oder zum Katalog-Index, und der Core-Entwickler hat gesagt, er macht nur das CMS. Es gibt also selbst im Kern-Team Leute, die die Software bauen, sich aber nur mit einzelnen Komponenten auseinandersetzen, weil die Software so komplex ist. Man kann nicht alles von Magento kennen, es ist definitiv so, dass auch in Entwicklung sich erst mal in den Überblick verschaffen muss und sich in ein Thema einarbeiten muss wenn es um Anforderung geht.

Frage: Wie läuft denn das z.B. mit der Einarbeitung in ein neues Thema? Angenommen ich bekomme eine User Story, wo ich weiß, es ist ein Teil von Magento mit dem ich mich noch nicht befasst habe und weiß nicht, wie ich da vorgehen werde. Ich muss aber zunächst mal eine Schätzung abgeben.

Die Schätzung ist in dem Moment wo ich mich mit der Komponente nicht so gut auskenne wahrscheinlich noch größer als wenn ich mich damit auskenne. Wenn man Scrum macht und das nur abstrakt in Story Points angeben muss, ist es nicht so das Problem. Wenn man dem Kunden aber in konkreten Ansagen in Tagen machen muss, dann müssen wir schon in gewisser Weise in Vorlistung gehen indem wir die Story schätzen und uns dabei schon ein bisschen in die Komponente einarbeiten, um in den besten Überblick zu haben.

Wir gehen dann lieber das Risiko ein, dass der Kunde es im Endeffekt doch nicht beauftragt, als den Auftrag durch ein zu große Schätzung zu gefährden. Aber anders geht es nicht, weil

das System einfach zu groß geworden ist.

Frage: Kommt das noch regelmäßig vor oder wird es mit der Zeit weniger?

Mit der Zeit wird es weniger, aber es passiert gerade wenn man neue Entwickler hat, die sich noch nicht gut damit auskennen und dann etwas schätzen sollen, auch die Erfahrung des Schätzers machen sollen. Dem sollte unser Mönch nach immer der Schätzer, der es macht. Und wenn der, der es vielleicht besser könnte, nicht kann, dann muss sich der neue Entwickler beinarbeiten. Man will ja das Level des Teams gleichmäßig hochziehen.

Thema: PHP-Frameworks

Frage: Es ist einfach, Business Logic im Template auszuführen. Ist die Trennung zwischen Template und Programm also nicht stark genug gegeben oder einfach nicht forciert?

Wenn man sich den Code anschaut, wird es vorgegeben aber nicht forciert. In anderen Frameworks gibt es Template Engines mit eigener Syntax wie Fluid in Flow3. Da funktioniert in Plain PHP im Template, man kommt also gar nicht erst in die Versuchung, Business Logic im Template zu schreiben. Magento ist da nicht strikt genug, ich ziehe es aber vor, Plain PHP im Template zu schreiben, ohne zusätzlich Template Syntax

Frage: Jetzt erlaubt Symfony z.B. Plain PHP Templates, ohne dass man von dort auf alles zugreifen kann, da alle Abhängigkeiten von Anfang an sehr klar festgelegt werden, wohingegen man in Magento von jeder Stelle aus über Mage::pp() an alle Bestandteile heran kommt. Ist das vielleicht das Problem, also dass es zu viele Abkürzungen gibt?

Ich glaube, in dem Moment wo ich PHP zur Verfügung habe, kann man solche Einschränkungen immer irgendwo umgehen. In dem Moment, wo ich die Freiheit habe, gibt es immer Leute, die die Freiheit nutzen, weil sie es nicht besser wissen. Und das ist die größte Herausforderung dabei, dass ich mich als Entwickler an die Vorgaben des Frameworks halten muss. Siewird nicht erzwungen, wie bei Flow3, was ich persönlich aber als Risiko-Vorteil sehe. Ich muss mich dran halten, klar, aber ich habe alle Freiheiten, wenn ich sie mal braucht. Das ist für mich der größte Unterschied zwischen Zend Framework und Flow3, einmal die strikte Framework wo ich nur noch Klassen schreiben und Objekte, die miteinander agieren und einmal Zend Framework, aus dem ich einfach Komponenten nutzen kann, aber nicht in irgendwas erdrängt werde, in dem ich mich gar nicht aufhalten möchte.

Frage: Flow3 bringt also mit der eigenen Template-Syntax noch eine Abstraktionsebene hinein, ansonsten ist es bei reinen PHP-Frameworks aber so, dass man immer Abkürzungen finden kann, um irgendwie unsauber zu arbeiten.

Ich sehe das nicht als Nachteil. Ich sehe es nicht als Aufgabe des Frameworks, den Entwickler zu zwingen sondern der Entwickler soll sich seiner Verantwortung bewusst sein, qualitativ hochwertige Software zu schreiben. Das heißt die Rollenvertauscht, wenn das Framework versucht, den Entwickler dazu zu zwingen, sauberen Code zu schreiben. Wenn der Entwickler diesen Anspruch gar nicht hat, dann wird das auch im Framework nicht schaffen.

Thema: Einstieg

Frage: Nun ist es gerade bei PHP eher umgekehrt, dass es einen nicht nur nicht dazu erzieht, sauberen Code zu schreiben sondern eher dazu verleitet, auch

mal unsauberen Code zu schreiben. Das ist bei Plain PHP oder älteren PHP-Anwendungen wahrscheinlich stärker der Fall als bei Magento aber merkt man das noch in der Entwicklung?

Bei uns ist es so, dass nur Leute mit ihrem Plain PHP Wissen kommen und der Code total krautig ist. Man muss, wenn man in die Entwicklung muss, auch hier das Potential sehen, sauberen Code zu schreiben als das was momentan schreibt. Er muss hier den Anspruch haben, strukturiert zu arbeiten als direkt von Anfang an strukturiert zu arbeiten, wie ich dir sagst, PHP ist so einfach macht, unsauber zu schreiben. Aber wirst du schon sehr viel Energie in den Programmen sauberer Arbeitswiese bringst, und dann ist es auch möglich. Aber es ist halt immer so, dass man sich selbst dazu bringen muss, man muss selbst diesen Anspruch haben.

Ich komme selbst hier aus der C# Ecke, hab früher mit .NET programmiert, deshalb hab ich mich auch in Magento oder Zend Framework wohl gefühlt als in irgendwelchen Typo3 oder alten PHP 4 Anwendungen. Klar ist es erstmal merkwürdig, dass die Variablenkennungen Typ haben, daran muss man sich als Programmierer erstmal gewöhnen, wenn man vorher mit Java oder C# gearbeitet hat, also daran dass so in Variablen Typ springt oder springen kann wenn man nicht aufpasst. Ich würde mit PHP wahrscheinlich auch kein Raketensoftware programmieren. Es ist genauso wie du sagst, dass es PHP immer nicht schwer macht, krautig zu programmieren.

Frage: Jetzt habt ihr also die typische Situation, dass ihr neue Programmierer bekommt, die eher aus der PHP Ecke kommen und die anlernen müsst, sauber zu arbeiten. Geht das alles über den Review-Prozess oder anfängliches Training oder wie läuft das?

Der asynchrone Review-Prozess macht später Sinn, wenn in gewisser Level erreicht ist. Am Anfang muss man noch persönlich als Mentor dabei sitzen, zeigen, wie gut der Code aussieht, und die Entwickler dazu bringen, z.B. Magento-Code zu lesen, zu schauen, wie die Controller das machen. Auch da muss man darauf hinweisen, dass es teilweise nicht so schön programmiert ist.

Und mit schließlich Businessplan kann man auch viel erreichen. Wir haben ja schon einige Projekte, aus denen wir auch Navigativ-Businessplan zeigen können, und was die Auswirkungen sind. Es wurde z.B. in einem Login-Template in SOAP-Client integriert, mit dem die Magento-API des Shops angesprochen wird, in dem man sich gerade befindet. Über die API wurden alle Customergänge geholt und dann Zettel für Zettel das Kennwort verglichen. So etwas bleibt dem Entwickler dann auch im Kopf, und er sagt, so will ich nicht werden.

Oft ist es ja so, dass gefragt wird, „Warum soll ich denn nicht im Template Business Logic schreiben?“, und da sieht man mal, was passiert, wenn man das schliefen lässt. Ich glaube, diese Navigativbusinesspläne sind ganz gut.

Thema: Verbesserungs-Potential

Frage: Zusammengefasst, wo siehst du am ehesten Verbesserungs-Bedarf und -potential im gesamten Entwicklungsprozess?

Einmal sehe ich Verbesserungspotential bei der Schulung des Kunden – oder des Product Owners, der bei uns oft auf Kundensitzungen und in wichtiger Rolle im Scrum Prozess einnimmt. Man möchte den Kunden in den Prozess einbeziehen, dazu muss ihm aber überhaupt bewusst

s in, welche Rolle, welche Verantwortung er im Entwicklungsprozess hat. Das ist bei jedem Projekt auf jeden Fall Herausforderung und in Punkt, der verabschiedet werden muss und an dem man auch dran bleiben muss. Das bezieht sich auch darauf, dass die Anforderung in Form von User Stories präziser und Ablauforientierter werden.

Dann ist das größte Thema das automatisierte Testen, was gerade bei agiler Entwicklung sehr wichtig ist. Es gibt da schon in Magento, was man machen kann aber wir nutzen es noch nicht, es ist in unseren Prozess noch nicht integriert. Es ist aber auch mit Magento momentan nicht in vollem Umfang möglich.

Frage: In Bezug auf Magento: Wo könnte man beim Framework selbst ansetzen, um die Arbeit damit noch zu erleichtern?

Es wäre super, wenn es von Haus aus Unit-Test wäre. Es würde uns enorm helfen, wenn wir dann nur noch die richtigen Extensions testen müssen. Dieses Test Automation Framework habe ich mir noch nicht angeschaut aber das geht zumindest in die richtige Richtung.

Die Architektur finde ich schon gut und reviewbar gelöst, es gibt mit dem Event-System Möglichkeiten, in dem Core einzugreifen, ohne den Core zu überschreiben. Es gibt andere Möglichkeiten, den Core zu erweitern. In dem Sinn bin ich da schon mit zufrieden mit der Grundlage auf der man Shopsystem baut vs. komplette Neuanfangen.

Einzelne Komponenten sind nicht so flexibel sind wie man sie gerne hätte. Das PDF-Modul ließ sich zum Beispiel mit DomPdf besser lösen. Es ist aber auch in Einschränkung, dass so in Framework den Anspruch hat, auf möglichst vielen Plattformen zu laufen, also von möglichst wenigen externen Bibliotheken abhängig zu sein, die der Host installieren muss. Das sind Entscheidungen, die so in Architektur auch beeinflusst, da kann man den Magento-Ingénieurur nicht so böses in.

An vielen Stellen fehlenden allrdings Events und da ist schade, dass wenn man Verbesserungsvorschläge an Magento schickt, die nicht schnell genug im Core landen. Denn irgendwo in Event zusätzlich zu werden ist glaube ich nicht so schlecht für den Core, macht nichts kaputt, würde aber viel weniger Aufwand sparen.

A.3 Sonia Wrodarczyk

Datum: 23. Nov. 2012, Dauer: 1,5 Std.

A.3.1 Kurzfragebogen

Nam : Sonia Wrodarczyk
Firma: cod 4busin ss Softwar GmbH
Position: Proj ktl itung, Quality Assuranc
Entwickl r-T am-Größ : 3-4 Entwickl r
Mag nto-Erfahrung: 1-2 Jahr
V rw nd t s Proz ssmod ll: XP (in Ansätz n)

A.3.2 Transkription

Fabian Schmengler: Ihr v rsucht, XP anzuw nd n, wi sind da di Erfahrung n, wi w it funktioni rt das hi r?

Sonia Wrodarczyk: Was wir von XP ins tz n ist d r Proz ss wi wir di Aufgab n im Proj kt int il n. Wir hab n in M ng Aufgab n, nutz n in Onlin -Tracking Syst m (Jira) und hab n dazu in Taf l im Büro wo di Aufgab n aufg t ilt sind in „op n“, „todo this w k“, „in progr ss“, „t sting“, „r vi w“ und „don “.

W nn di T sts durchlauf n sind, wird von Nikolai [Krambrock, G schäftsführ r] d r Qu llcod durchg s h n und dann komm n di Tasks in „don “.

Das sind di Spalt n im Board und j d r Mitarb it r hat in Z il .

FS: Das sind also di Phas n. Funktion i r n manch Phas n b ss r als and r ?

SW: Was oft zu s h n ist, b sond rs b i Mitarb it rn mit w nig Erfahrung, ist dass di Aufgab n lang brauch n bis si f rtig sind, w il in d r T st-Phas F hl r g fund n w rd n, und di Aufgab zurück auf „in progr ss“ g ht. Es gibt vi l It ration n, wo s imm r wi d r zurück g ht, oft g ht in Task zw i od r dr i mal gar nicht rst bis zum R vi w, sond rn vom T st n imm r wi d r zurück.

FS: Gibt s k in T sts, di di Programmi r r s lb r durchführ n?

SW: Doch. Es ist T il d r Anford rung, dass di Programmi r r ihr Sach n rst s lbst t st n. Ab r si hab n wahrsch inlich in and r P rsp ktiv b im T st n und find n di Sach n nicht, di in normal r Nutz r, d r k in Ahnung vom Programmi r n hat, find t.

FS: Ab r s w rd n schon auch T st-Tools wi S l nium währ nd d r Entwicklung ing s tzt od r kommt das rst zum Schluss?

SW: Also mit d n S l nium-T sts ist s so: di w rd n dann ntwick lt, w nn di Aufgab schon fast durch ist. Si w rd n als Üb rwachungs-Syst m g nutzt, um nachh r zu üb rprüf n, ob in d m Proj kt di Aufgab n, di schon f rtig sind und funktioni r n nicht durch Commits aus d n nächst n Aufgab n kaputt g gang n sind.

FS: Si w rd n also nachträglich für R gr ssionst st v rw nd t und nicht b i d r Entwicklung um zu s h n, ob di Anford rung rfüllt ist. Ich würd daraus schli ß n, dass das T st n währ nd d r Entwicklung noch in Probl mf ld ist. Dass da noch k in Syst m hint r st ckt, dass di Programmi r r mit ihr r ing schränk t n P rsp ktiv und unsyst matisch m T st n F hl r nicht find n. Kann man das so sag n?

SW: Ja.

FS: Gab s da mal V rsuch , das Probl m anzug h n und syst matisch r zu t st n?

SW: Was g macht wurd ist, m hr W rt darauf zu l g n, dass di Programmi r r rst s lbst t st n und da m hr Z it hin in inv sti r n, b vor si d n Task in di T st-Phas schi b n. Das Probl m ist auch, w nn j mand twas programmi rt hat, s in di T st-Phas schi bt und in d n rst n dr i, vi r Minut n schon in F hl r g fund n wird, dann v rschi bt sich das z itlich, d nn s dau rt natürlich twas, bis d r T st r d n T st auch durchführt. Das wird nicht sofort g macht, sond rn m ist inmal am Tag n hm ich all Aufgab n in „T sting“ und t st .

W nn also in Aufgab auf „T sting“ hängt, b schäftigt sich d r Programmi r r in d r Zwischenz it schon mit in r and r n Aufgab und dann kommt di alt Aufgab zurück. Dann hab n si m hr r angfang n Aufgab n und komm n auch nicht so schn ll zum End .

FS: Klar, da man sich imm r aufs N u inarb it n muss... In d n Phas n Programmi rung/T st b st ht also das Probl m, dass vi l Aufgab n wi d r zurück wand rn.

SW: Ja.

FS: Wi ist das mit d n T sts, di du manu ll machst. Probi rst du all Anw ndungsfäll inmal durch, hast du da in Syst m?

SW: Es gibt in paar Sach n, di man standardmäßig durchg ht. Such und Ch ckout sind di wichtigst n Funktion n in in m Onlin -Shop, also dass man Produkt sucht und dann kauft. W nn man Erw it rung n ntwick lt, di in b stimmt Fähigk it hinzufüg n, ori nti r ich mich natürlich daran, was g änd rt wurd . Abhängig davon an w lch r St ll das im Shop ist, v rsuch ich, zu üb rl g n, was da kaputt g h n könnt .

FS: Du t st st auch j d s Mal manu ll di Standard-Funktion n? Könn t das nicht in S l nium-T st automatisch mach n?

SW: M ist ns ist, w nn ich twas manu ll t st , noch k in S l nium T st vorhand n. Es kann s in, dass sich twas im Ch ckout g änd rt hat, zum B ispi l b i d r Adr ss od r b i Zahlungsdat n. Das sind auch Sach n, di man unt rschi dlich im Back nd inst ll n kann und nicht so infach mit S l nium zu t st n sind. D shalb g ht s in d r rst n Rund schn ll r, manu ll zu t st n. W nn man dann f stst llt, dass di Ding funktioni r n, hat man auch in Id , was man im S l nium-T st ins tz n könnt , d r dann imm r wi d r ausg führt wird.

FS: T st st du mit v rschi d n n Dat n v rschi d n Sond rfäll od r g hst du nur Standard-Fäll durch?

SW: Ich v rsuch schon, v rschi d n Fäll zu t st n. Standard-B ispi l Ch ckout: Da gibt s di Fäll „Eing loggt r Kund“, „Gast“ und „Kund d r sich g rad n u r gistri rt“. Oft ist s so, dass in Funktion b im Gast funktioni rt, b im ing loggt n Kund n ab r nicht m hr. D sw g n v rsuch ich das ntspr ch nd, mit v rschi d n n Fäll n zu t st n.

FS: Entsteht dadurch vielleicht die Wiederholung von Arbeit im Test, vielleicht was du doppelt machen musst?

SW: Manche Schritte werden sich natürlich. Meistens sind das nur zwei, drei Fälle. Es sind dann, es handelt sich um eine Eigenschaft, wo es wirklich unterschiedliche Fälle geben kann, und das besteht ist, repräsentativ Fälle für irgend eine Eigenschaft des Systems zu testen. Ich hatte schon einmal in einem Fall, wo ich mit mehreren Eingaben testen musste, da ging es um Steuerung. Das war zusätzlich von der Zahlungsart abhängig, deswegen musste man das alles durchgehen, das habe ich dann in einer Excel-Tabelle dokumentiert. Aber das sind nicht Sachen, die oft kommen.

FS: Das sind ja Fälle, die man eigentlich testen müsste. Kommt es denn vor, dass Sonderfälle übershnen werden, die dann später zum Fehler führt?

SW: Das passiert manchmal, es ist zum Beispiel passiert, dass ich keinen Fehler gefunden habe, und der Quellcode auch durch den ersten Review gegangen ist. Der Programmierer wurde gebeten, das nochmal durchzugehen und dann wurde der Fehler nicht im Frontend gefunden, sondern im Quellcode, wo es irgendwo in den Klassen Fehler gab. Das sind Sachen, die nicht so einfach zu finden sind. Das Problem mit Magento ist, dass oft irgendwelche ganz unerwartete Fehler auftauchen. Man hat etwas programmiert und damit in einer Funktion kaputt gemacht, aber nicht damit gearbeitet.

FS: Das müsstest du eigentlich die Selenium-Tests finden. Die können auch nicht alles abdecken?

SW: Ja, was man zum Beispiel mit Selenium nicht testen kann ist, ob das System Emails versendet. Es gibt zum Beispiel in Email nach dem Checkout oder nach irgendwelchen Formularen. Und Selenium kann sich nicht in den Mailbox einloggen und checken, ob die Email angekommen sind.

FS: Da würde also noch in Möglichkeit fehlen, E-Mails zu testen...

SW: Aber was ich gut finde ist, dass Selenium die Standard-Tests, die man immer wieder machen könnte, das Testen von einfachen Sachen überdeckt. Man kann so in den Test laufen lassen und muss die einfachen Sachen nicht mehr manuell testen.

FS: Würde es helfen, wenn es einfacher wäre, die Tests mit verschiedenen Daten zu füllen, dass alle Sonderfälle abgedeckt werden ohne dass man sie einzeln eingeben müsste?

SW: Die Frage ist, was ist aufwändiger: Die Sachen einmal manuell zu testen oder die Daten in Selenium zu geben und mit ihnen automatisch den Test laufen zu lassen. Was für mich mit Selenium problematisch ist, dass die Tests manchmal nicht sofort funktionieren und man noch Kommandos per Hand eingeben muss. Als in Person, die nicht programmieren kann, ist das für mich problematisch.

FS: Das heißt, manuelle Tests sind erstmal weniger aufwändig, auch wenn du in Sachen manchmal durchtestest.

SW: Für mich ist es manchmal so, dass ich mit einem Selenium-Test mehr Zeit verbringen als wenn ich das manuell testen will. Wenn es um einen Test geht, der nur einmal durchgeführt wird, um etwas zu überprüfen, dann ist es also in Ordnung wenn er nur manuell ist. Aber wenn es sich um Tests handelt, die immer wieder ausgeführt werden müssen, dann ist das besser – auch mit vielen Fällen – das von ihnen automatisch den Test machen zu lassen.

FS: Die Frage ist, wie wahrscheinlich ist es, dass ein Test nur einmal durchgeführt werden muss oder öfter? Was sind die für Tests die nur einmal durchgeführt werden müssen?

SW: Da muss ich überlegen... bei den letzten Projekten vermischt man wirklich in dem Moment in Fällen Selenium-Tests einzusetzen.

FS: Also im Prinzip sind automatisch Tests immer sinnvoll, auch wenn manuelle Tests nicht erst mal besser sind. Oder werden sie irgendwann zu viel, da sie doch recht lang brauchen, um durchzuführen?

SW: Man versucht schon, Tests nicht zu weit rholen, die genau das gleiche machen. Man versucht, in den Fällen in Tests zu behandeln, also dass nicht zu viel Eigenschaften pro Test getestet werden. Wenn ein Test fehlschlägt, können wir so schnell rfeuert werden, was genau das Problem ist. Aber man muss aufpassen, dass sich Tests nicht weit rholen, zum Beispiel genau das gleiche mit anderen Produkten. Damit das Sinn macht, müsste sich das Produkt schon wesentlich unterscheiden, zum Beispiel in einfachen Produkten und in konfigurierbaren, das ist schon ein wesentliches Unterscheid.

Wie lang so ein Test-Durchlauf ist, hängt davon ab, wie viele Tests es gibt, wie lang die einzelnen Tests sind und ob irgendwelche Fehler gefunden werden. Wenn das System Fehler findet, gibt es zum Beispiel mal in den Timeout von 3000 Sekunden [sic] und dann dauert das natürlich länger. Aber die Regel ist schon zwischen 5 und 8 Minuten bei über 40 Test Cases.

FS: Ist das ein Problem, dass der Durchlauf so lang dauert?

SW: Es ist ein Problem, wenn zu viel Leute an der Sache gleichzeitig arbeiten wollen. Es gibt zum Beispiel Sachen, die man manuell überprüfen möchte, aber wenn in Programmieren Änderungen hochspielt hat, werden die Selenium-Tests automatisch ausgeführt. Wenn ich gleichzeitig manuelle Tests in der Instanz teste und der Programmierer weiß das nicht, schlagen immer in paar der automatisch Tests fehl, die durch mein manuelle Tests gestört werden.

FS: Die Selenium-Tests müssen auf demselben System laufen, auf dem du manuelle teste?

SW: Ja. Jeder Programmierer entwickelt lokal auf seiner eigenen Instanz und speilt die fertigen Änderungen hoch. Ich teste auf der allgemeinen Instanz, weil ich mir da sicher sein kann, dass es die alleraktuellste Version ist.

FS: Das bringt uns zum Thema Integration. Die Entwickler entwickeln und testen auf ihrer Instanz und ja erst mal ihre eigenen Änderungen und erst auf dem Testsystem kommen alle aktuellen Änderungen zusammen, richtig?

SW: Nicht wirklich. Die Änderungen lokal werden zu erst heruntergeladen und dann neu entwickelt. Wenn natürlich in Programmieren etwas schnell rzu Ende gebracht hat, und das schon hochspielt hat, dann kann es sein dass der nächste in Änderungen hochspielt und es in den Konflikt gibt. Das kann schon passieren, wenn man hier gleichzeitig am selben Projekt arbeitet. Dann hat man die Daten nur bis zum Moment, wo man mit der eigenen Aufgabe angefangen hat. Es ist aber nicht so dass auf der letzten Magento-Instanz programmiert wird.

Dazu kommt dass wir die Templates nicht selbst entwickeln, das macht immer jemand anderes, oft passiert da die Kommunikation auf Kundensite. Das ist auch ein Problem in Integrationsproblemen, dass etwas nicht zusammen mit dem Template funktioniert oder das Template etwas kaputt macht oder etwas nicht berücksichtigt.

FS: Kannst du typisch Problem bei dieser Integration?

SW: Zum Beispiel haben wir im Modul Buttons, die im Tmplat nicht berücksichtigt werden, dafür hat das Tmplat aber eigene Buttons und das funktioniert nicht zusammen. Wenn man das merkt, muss man sich mit dem Tmplat abstimmen und erklären, dass sich das Tmplat mit den Buttons aus dem Modul verknüpfen. Was auch ein bisschen problematisch war, wenn die Programme r-r-twas machen, was vom Tmplat abhängig ist. Das versuchen wir zu vermeiden, die Module müssen selbständig funktionieren aber manchmal vergisst man das und das ist problematisch.

FS: Also es ist oft einfacher, etwas vom Tmplat abhängig zu machen aber sobald sich das Tmplat zum Beispiel ändert, gibt es Probleme.

SW: Genau.

FS: Funktioniert es denn, wenn man darauf achtet, die Module immer unabhängig von dem Tmplat zu machen?

SW: Ja, denn wenn etwas nicht funktioniert, kann man erstmal das Tmplat abschalten und gucken, ob es ohne das Tmplat funktioniert. Wenn ja, ist es wahrscheinlich ein Problem im Tmplat.

FS: Das ist auf jeden Fall ein Vorteil, wenn man schafft, die Module unabhängig vom Tmplat zu stellen. Die Frage ist, schafft man das auch immer?

SW: Nicht immer, aber es ist jetzt schon ein klarer Anforderungsbereich.

FS: Wie ist das mit den Modulen untereinander? Gibt es da Konflikte?

SW: Ja. Wir versuchen es so zu machen, dass die neuen Sachen die man programmiert, mit den Sachen zusammen funktionieren, die bereits integriert sind. Aber was zum Beispiel passiert oder passieren kann, wenn man jetzt ein Modul installiert, ist: wir haben ein Modul programmiert und dann in Zahlungsmodul installiert und dann kracht's, weil die Module die gleichen Klassen überschreiben. Dann muss man schon nachträglich das eigene Modul anpassen, ansonsten kann man nichts ändern.

FS: Sind für Module in großen Problemen? Nutzt ihr überhaupt viel für Module?

SW: Probleme gibt es da nicht so oft aber wir nutzen auch nicht viel. Nur bei Sachen wie Bonitätsprüfung, Zahlungsanbietern, das sind Sachen die man eigentlich nicht anders machen kann.

FS: Zur Anforderungsanalyse, bist du da mit involviert?

SW: Bis jetzt war ich noch nicht dabei, wenn die Anforderungen formuliert werden. Die Aufgaben kommen schon fertig bei mir an, ich bin dafür verantwortlich, die Aufgaben für die Entwickler zu übersetzen. Dazu muss ich schon verstehen, worum es geht und fähig sein, die Sachen zu erklären.

FS: Es geht also nicht nur darum, sich von Deutsch nach Englisch zu übersetzen, sondern auch sozusagen von Kundensprache nach Entwicklungssprache?

SW: Richtig in „Entwicklungssprache“ kann ich sie natürlich nicht übersetzen. Aber ich sehe, dass die Entwickler ein bisschen in die Dinkis haben, und da muss ich schon mal erklären, dass etwas so und so aussieht muss damit die Funktionalität einfach logisch ist.

Nicht nur, weil es einfach zu entwickeln ist, wir möchten allgemein nicht zu komplizierten Lösungen inszenieren, damit sie nachvollziehbar sind. Andererseits soll man die Funktionalität nicht zu sehr vereinfachen. Es muss logisch sein, was da passiert.

FS: Du sagst, die Entwickler haben ihr Design, worauf musst du dann achten, wenn du ihn nicht weiter erklärst?

SW: Oft ist es so, dass Fragen kommen, zu denen es eigentlich nur eine einzige Lösung gibt, wenn man ein bisschen darüber nachdenkt, weil die andere unpraktisch ist oder nicht ganz logisch.

FS: Das heißt, die Entwickler fählt die Sicht des Anwenders, darauf was praktisch ist?

SW: Ja, also es ist nicht immer so oder nicht so drastisch aber ich habe manchmal den Eindruck dass sie es nur von der Perspektive sehen, wie es technisch umgesetzt wird. Sie machen zwar das, was im Task steht, aber ohne sich Gedanken zu machen, wie das von der Perspektive der Kunden aussieht, was für ihn einfach zu benutzen unpraktisch ist, oder warum jetzt etwas im Task steht.

FS: Du musst also die Aufgaben noch genauer erklären, die Logik erklären, Dinge die vielleicht aus Kundensicht selbstverständlich sind.

SW: Ja. Und es ist oft so, dass der Kunde nicht ganz so technisches Wissen hat. Deswegen gibt es Sachen, die nicht hundertprozentig geklärt sind, wo man sich denkt, das könnte man auf zwei oder drei Weisen machen.

FS: Kann es auch sein, dass in Aufgaben auch auf mehrer Weise vorgegangen werden kann, also was überhaupt gemeint ist, unabhängig von der technischen Umsetzung? Wie geht ihr damit um, wenn Aufgaben vorschidenintentional werden können?

SW: Ja, die Entscheidungen sind so: Wenn dem Programmierer nicht klar ist, worum es geht, wird das erst mit mir abgestimmt und ich überlege, wie man das logisch machen könnte. Wenn es dann wirklich mehrer Möglichkeiten gibt, und es weiterhin nicht klar ist, wird das mit Nikolai abgestimmt, der den direkten Kontakt mit dem Kunden hatte. Vielleicht wurde irgendwo im Gespräch gesagt und nicht aufgeschrieben, oder rückt die Design des Kunden beiseite. Er kann nicht alle Details genau aufschreiben, irgendwas ist immer nicht im Text. Das muss dann von Nikolai entschieden werden, da ist dann das Problem dass es immer etwas länger dauert bis die Entscheidung getroffen wurde.

FS: Geht es auch so weit dass beim Kunden zurückgefragt werden muss?

SW: Es kommt schon vor, dass etwas so unklar ist, aber das wird dann erst gesammelt, damit es nicht zu vielen Rückfragen gibt.

FS: Gibt es abgesehen davon noch Probleme dadurch? Kommt es zum Beispiel auch vor, dass der Kunde am Ende doch etwas anders wollte?

SW: Manchmal, aber wir versuchen schon darauf zu achten, dass das vorher geklärt wird und nicht am Ende. Es ist natürlich viel Arbeit, wenn man die Sachen von neuem anfangen muss.

FS: Jetzt bist du auch ein bisschen mit, was die technische Umsetzung angeht. Die Entwickler müssen zunächst mal entscheiden, was es für technische Möglichkeiten gibt und

verschiedenen Entwürfe machen, bis sie in den Grundlagen haben, die funktionieren. Hast du da ein bisschen Einblick?

SW: Nicht immer. Manchmal ist es so, dass mehrere technische Lösungen vorhanden sind, dann überlegt sich der Entwickler, was er sagen will und dann diskutieren wir ein bisschen, was so die Unterschiede, die Vor- und Nachteile der verschiedenen Lösungen sind. Wenn ich den Eindruck habe, dass ein von den Möglichkeiten besser ist, kann ich auch ein bisschen Anweisung geben. Aber das ist nicht so oft der Fall, weil viele von den Sachen mir technisch unklar sind und dann stimmen die Entwickler das hier mit Nikolai ab.

FS: Das wäre die nächste Frage gewesen, liegt es dann in der Entscheidung der Entwickler? Du sagst, es wird mit Nikolai abgestimmt.

SW: Also, es ist schon von seiner Seite mehr das Ziel, dass die Entwickler mehr Sachen selbstentschieden können und in der Angabe machen, was die beste Lösung ist. Damit auch nicht bei der Aufgabe immer wieder in der kommt und fragt, „was soll ich das machen“, sondern dass sie imstande sind, selbständig zu arbeiten.

FS: nun kann das nicht jederzeit von Anfang an, und nicht jederzeit alle Möglichkeiten kennen. Wie ist das, was für Hinweise dann von Nikolai kommen? Wird da jemand in der Lösung gesucht, gibt es Tipps für die Entwickler oder wie wird das abgestimmt?

SW: Solche Sachen kommen oft bei den Team Meetings. Die gewünschte Vorgehensweise ist so, dass die Entwickler in der Problemdarlegung, was zu tun ist, und fertige Vorschläge wie man das umsetzen kann und dann wird diskutiert und die beste Lösung ausgewählt. Es soll nicht so ablaufen: „Das ist zu tun, ich habe keine Ahnung, was es macht werden soll, was soll ich das machen?“ sondern: „Ich habe mir das überlegt, man könnte es so, so oder so machen, je nachdem die Umsetzungsmöglichkeiten hat die und die Vor- und Nachteile, die Frage ist, welche von den Möglichkeiten soll ich auswählen?“

FS: Man könnte also sagen, dass die Entwickler, wenn sie nicht ohnehin schon genau wissen was sie machen müssen, erstmal verschiedene Möglichkeiten suchen und verschiedene Entwürfe machen. Dazu versuchen sie, die Vor- und Nachteile herauszubekommen, und das wird dann im Team Meeting diskutiert.

SW: Ja. Manchmal ist es so, dass wenn der Programmierer sich besser überlegt, was für Möglichkeiten es gibt, realisiert dass eine Lösung gar nicht geht und nur noch einbleibt. Dann kann er selbstentschieden, dass es so geht. Im Team Meeting wird dann einfach bestätigt, was er grad macht.

FS: Also hilft diese Art Entwurf schon oft, herauszufinden, was funktionieren und was nicht bevor man anfängt etwas zu programmieren was vielleicht gar nicht funktionieren kann.

SW: Ja.

FS: Wird das denn auch immer gemacht?

SW: Ich bin mir nicht sicher, ob das wirklich in jedem Fall passiert. Wenn die Programmierer die Aufgaben bekommen, dann sollen sie selbst schätzen, was viel Zeit sie brauchen. Was dort passieren soll ist, dass sie sich überlegen, was zu tun ist und wie man es vielleicht umsetzen könnte.

FS: Bevor oder nachdem sie die Schätzung abgegeben?

SW: Das soll n sich übrl g n, b vor si di Schätzung abg b n. Ab r ich hab d n Eindruck, das passi rt nicht imm r so ganz g nau, sond rn rst danach, w nn si di Aufgab schon auf d m Board hab n. Dann such n si Lösung n und hab n manchmal in Id , dass s nur mit in r b stimmt n Ums tzungsw is g h n kann und manchmal übrl g n si sich m hr r Möglichk it n.

FS: Dein Eindruck ist, dass si sich das oft rst anschau n und übrl g n, w nn si d n Task anfang n und nicht w nn si ihn das rst mal anschau n und schätz n?

SW: Ja. Di Schätzung passi rt manchmal zi mlich schn ll, d shalb d nk ich mir, da könn n si noch gar k in Lösung wiss n, vi ll icht ist das nicht so g nau.

FS: Di Schätzung wird in Stund n od r Tag n abg g b n?

SW: Abhängig von d r Aufgab inig Stund n od r inig Tag ab r schon in Zit.

FS: Ist das d nn inig rmaß n g nau? Du sagst, si g b n di Schätzung n oft s hr schn ll ab, also ist das in Probl m?

SW: Das passi rt vi ll icht auch in bissch n d sw g n, w il si m hr r Aufgab n zug wi s n hab n und si nicht all inn rhalb d r aktu ll n Woch mach n. Also si soll n auch di off n n schon schätz n und davon sind dann nicht all auf „Todo this w k“ g s tzt. Vi ll icht passi rt das so schn ll, w il di Aufgab n schon vorh r g schätzt w rd n. Di Aufgab n, di angfang n wurd n ab r nicht in d r vorg s h n n Schätzung zu End g bracht w rd n konnt n, w rd n umg schätzt. Da sind dann irg ndw leh Probl m aufg tr t n. Das ist so in bissch n das Probl m, dass sich di Schätzung n in d r Zwisch nz it änd rn.

FS: Wann wird das d nn korrigi rt? W rd n j d Woch di aktu ll in B arb itung b findlich n Aufgab n n u g schätzt?

SW: Ja, s ist oft so, dass in Programmi r r für in Aufgab ursprünglich zw i Tag g -schätzt hat und w nn r mit d r Aufgab angfang n hat, m rkt r, dass r noch in n Tag zusätzlich braucht.

FS: Und gibt s da in n f st n Plan, wann di s Schätzung korrigi rt wird? Od r wird si korrigi rt, sobald d r Programmi r r m rkt, dass r s nicht schafft?

SW: Erst vor d r n u n It ration, also vor d m Tam M ting, w nn di Aufgab n am Board diskuti rt w rd n und Aufgab n n u zug wi s n w rd n

FS: Würd st du sag n, dass di zw it , korrigi rt Schätzung nachd m d r Programmi r r mit d r Aufgab angfang n hat, g nau r ist od r wird da auch noch oft nach-korrigi rt?

SW: Unt rschi dlich. Manchmal r icht schon di zw it Schätzung ab r w nn s in ganz komplizi rt Aufgab gibt, passi rt s auch, dass si zw i od r dr i mal n u g schätzt wird. Das ist natürlich aus d r Manag m nt-P rsp ktiv nicht gut, w il man nicht klar sag n kann, wann wir mit d m Proj kt übrhaupt f rtig sind.

FS: Also di s Schätzung nach Zit ist in groß s Probl m, w il si s hr ung nau ist und imm r wi d r g änd rt w rd n muss.

SW: Ein and r Sach di kommt ist, di Programmi r r schätz n di Aufgab n s lbst, dann häng n di Aufgab n am Board und währ nd d s Tam M tings guckt sich Nikolai di Aufgab n an und di Aufgab n für di komm nd Woch w rd n zug wi s n. Dab i kommt imm r di Frag „Bist du dir sich r, dass du di Aufgab in vi r Stund n f rtig mach n kannst,

wi du das g schätzt hast? Du musst b rücksichtig n, dass du si noch zum T st n gibst und s s in kann, dass noch twas v rb ss rt w rd n muss!“ – das kommt schon öft rs – „Dann kommt s durch di T sts, s gibt k in F hl r m hr, ab r dann kommt d r Qu llcod zum R vi w und wir st ll n f st, da ist twas nicht so ganz schön, das muss and rs g schri b n w rd n und w nn das f rtig ist muss noch inmal g t st t w rd n, ob di Qu llcod -Änd rung nicht twas kaputt g macht hat. Bist du *sicher*, dass du di Aufgab in vi r Stund n f rtig mach n kannst?“ – „Ähh, nicht wirklich.“

Und so st llt man schon währ nd d s T am M tings f st, dass di Schätzung n nicht so g nau sind. Das Probl m ist, di s Frag kommt ig ntlich j d Woch , und di Programmi r r m rk n, dass si doch nicht so sich r sind mit ihr r g schätzt n Z it.

FS: Ab r di g schätzt Z it soll schon T st und R vi w mit b rücksichtig n.

SW: Ja, s soll di Z it, in d r di Aufgab f rtig ist, g schätzt w rd n. Also ig ntlich auch T st und R vi w s lbst, ab r m ist ns ist d r Ant il d r T st-Z it an d r Z it, di allg m in für di Aufgab b nötigt wird nicht so groß, s s i d nn s gibt wirklich vil F hl r und man muss m hrmals t st n. D r Entwickl r soll sich b im Schätz n schon auf s in ig n Tätigk it konz ntri r n, ab r b rücksichtig n, dass di Sach n zurückkomm n und v rb ss rt w rd n müss n. Es g ht vor all m um di s Nachb ss rung n.

FS: Vi ll icht noch in bissch n m hr zum T am M ting: Was ist da noch all s B standt il, wi soll das th or tisch ablauf n, was gibt s für V rb ss rungsmöglichk it n?

SW: Noch vor d m T am M ting w rd n di Z tt l mit d n Aufgab n n u g druckt, mit d n aktu ll n Schätzung n, und dann w chs ln di Programmi r r für di ig n n Aufgab n am Board di Z tt l. Im T am M ting ist d r V rlauf m ist so, dass d r Programmi r r sagt, was r grob g macht hat, also mit di s n Aufgab n ist r durch, di s hängt noch im T sting, warum hängt si da, warum ist di and r noch nicht durch, sond rn „in progr ss“? Es wird g klärt, warum twas nicht in d r vorg s h n n Z it g schafft wurd , und ob s irg ndw leh Probl m gab. Dann w rd n für di P rson n n u Aufgab n zug wi s n und dann ist auch Z it für di Frag n, di di P rson n zu d n n u n Aufgab n hab n. Si hab n sich zum B ispi l üb rl gt, wi si twas ums tzt n woll n und hab n di rst n Frag n.

FS: Das sind di Frag n an dich, üb r di du vorhin g sproch n hast?

SW: Di Frag n an mich komm n schon vorh r, w nn si di Aufgab n das rst mal l s n. Ob si di Aufgab n v rst h n, soll vorh r g klärt w rd n, si b komm n di Aufgab n ja vor d m T am M ting. Im T am M ting soll n di t chnisch n Frag n komm n, di Nikolai b antwort t.

FS: Wi ist das b i Probl m n mit d n aktu ll n aufgab n, du sagst s wird zum B ispi l darüb r g r d t, warum twas noch im T st hängt. Wird dann im T am M ting schon v rsucht möglich Lösung n zu find n?

SW: Ja.

FS: Und da ist j d r b t ilt, d m twas dazu infällt?

SW: W nn s aktu ll Probl m gibt, wird das im T am m ist ns schon vorh r abg stimmt. Ob zum B ispi l in and r r Programmi r r Ahnung hätt , wi man twas lös n kann. Da fragt man schon dir kt, und di Programmi r r stimm n sich mit inand r ab. Od r w nn s darum g ht, dass in d r Aufgab twas nicht so ganz klar war, dann kommt di Frag an

mich. Während des Teams Meetings kommen die Fragen, die dir in der Antwort kommen, die ich an Nikolai.

FS: Also alles was man so unterinandern klären könnte, ohne dass man über Nikolai oder den Kunden geht, wird dir jetzt geklärt. Das heißt, während der Entwicklung die ganze Woche über sprechen sich die Entwickler laufend ab?

SW: Ja. Und dann gibt es nach dem Team Meeting wo die Aufgaben zugewiesen werden noch in anderen Teams Meeting in den nächsten zwei Tagen, was kürzer ist (30 Minuten), wo noch irgendwelche Fragen kommen, die nicht selbstständig gelöst werden können. Nicht dass in Aufgaben bis zum nächsten Meeting in der Woche wartet.

FS: Zu dieser Absprache unterinandern: Zum XP gehört ja eigentlich noch Pair Programming – wird oder wurde das hier auch gemacht?

SW: Im Moment ist es meistens so, dass jeder selbstständig arbeitet, aber die die wenigere Erfahrung haben, arbeiten am Anfang mit jemand anderem zusammen. Wenn es irgendwelche Probleme gibt ist es auch manchmal so, dass die Programmierer sich zusammensetzen und gemeinsam Lösungen finden, aber es ist nicht Standard dass jeder Aufgabe gemeinsam gemacht wird.

FS: Also nur wenn jemand ganz neu ist oder nicht weiter kommt?

SW: Ja, nur in den besten Fällen.

FS: Woran liegt das, dass das nicht konsolidiert gemacht wird? Gehst du so schnell?

SW: Was schnell geht, bin ich nicht sicher. Pair Programming machen wir halt nicht durchgängig, daher ist das schwer zu schätzen. Aber wenn sich zwei bei Problemen zusammensetzen, sind sie schon schnell damit, in Lösung zu finden. Zumindest habe ich diesen Eindruck. Es kann auch daran liegen, dass wir nicht so viele Mitarbeiter haben und dann ist es vielleicht doch schnell, wenn sie getrennt Aufgaben machen. Das kann ich aber wie gesagt nicht hundertprozentig schätzen. Ich weiß nicht, wie man programmiert und wie die Leute arbeiten. Es gibt ja Leute, die besser arbeiten wenn sie selbstständig arbeiten und solche die in der Gruppe besser arbeiten.

FS: Du kannst also nicht konkret sagen, warum ihr euch entschlossen habt, das nicht durchgängig zu machen? Würde das Vier-Augen-Prinzip nicht helfen, mehr Fehler im Voraus zu finden?

SW: Schwer zu sagen. Wir haben es nicht ausgetestet, aber das kann gut sein, wenn in Person auf etwas nicht achtet dass der andere dann genau das auffällt. Es fehlt nur bisher die Möglichkeit, das auszuprobieren.

FS: B kommst du mit, ob es Probleme bei programmierten gibt, die immer wieder auftreten?

SW: Nicht wirklich. Was schon manchmal ist, dass der Server auf dem jemand programmiert zu langsam ist, dass sie sich über die Wartzeiten beschweren, dass etwas nicht schnell genug läuft.

FS: Das Team ist sehr klein, würde nicht stimmen Sachen in immer größeren Unternehmen besser funktionieren?

SW: Ich glaube, ja. Wir haben halt nicht so viele Mitarbeiter und wenn es in Projekt gibt, das in der Firma stimmt, dann ist abgeschlossen werden muss und wenn zu viele Sachen da lä-

ger brauchen, weil die Aufgaben zum Beispiel immer vom Testen zurück kommen, dann ist der Moment irgendwann da, dass Zieldruck kommt weil die Aufgaben zu irgendwann im Zeitpunkt abgegeben werden müssen. Wenn es mehr Leute gibt, kann man das besser auslagern, mehr Aufgaben gleichzeitig verteilbar, dann ist die Chance größer, dass man schon länger vor dem Abgabetermin fertig ist, in Ruhe durchtesten kann, und dem Kunden das Ergebnis hervorstellen kann. Dann ist nicht so viel Stress in der letzten Phase.

FS: Also mehr Programmierer um mehr Aufgaben gleichzeitig zu arbeiten... Zieldruck und Deadlines sind in der gesamten Thema. Habt ihr das tatsächlich in jedem Projekt, dass es zu einem bestimmten Zeitpunkt fertig sein muss?

SW: Nein, nicht in jedem Projekt. Also es gibt Projekte, wo zum Beispiel noch kein Online-Shop vorhanden ist und der Kunde möchte den Shop zu einem bestimmten Zeitpunkt online haben, dann ist der Zieldruck natürlich vorhanden.

FS: Ist das denn in Termin, den ihr auch dann auch verpflichtet, einzuhalten?

SW: Wenn es nicht fertig ist, kann der Shop natürlich nicht online gehen. Aber man versucht das schon so zu Ende zu machen.

Aber es gibt auch Projekte, bei denen schon ein Shop besteht und der Kunde möchte da nur zusätzlich Sachen haben. Da ist das nicht so streng zeitlich begründet, sondern nur grob, etwa „inhalb von einem Monat“ – das wird natürlich zwischen dem Kunden und Nikolai abgestimmt, auch je nach der aktuellen Kapazität.

FS: Ich frage, weil ihr ja eigentlich in agiles Modell fahren wollt, bei dem solche festen Zeitpläne nicht vorgeschrieben sind. Gibt es denn auch Projekte bei denen das anders abläuft?

SW: Ja, wir haben auch solche Projekte, wo etwas zu tun ist und wir machen in Version, wenn die fertig ist wird sie dem Kunden vorgestellt, dann sagt er „OK, also wir hätten hier gerne in zusätzliches Feld und das da nicht“, also möchte es noch ein bisschen anders haben. Solche Projekte haben wir auch.

FS: Also wo gar nicht von Anfang an alle Anforderungen klar sind, sondern sie immer wieder verfeinert werden, kann man das so sagen?

SW: Wir versuchen schon, die Anforderungen immer klar zu bekommen. Aber es gibt auch solche Fälle, wo der Kunde halt sagt, „Ja, das hier würde ich gerne anders haben“ aber das Projekt wird dann auch nach Aufwand berechnet und das ist dann in Ordnung. Wenn wir in Projekt haben, wo es in Deadlines gibt und was pauschal abgerechnet wird und sich der Kunde da jedes Wochenende ausdenkt, kann das Projekt nicht in der vorgeschriebenen Zeit enden.

FS: Ihr habt also beide Fälle, Projekte die nach Aufwand abgerechnet werden und wo auch immer Änderungswunsch kommen können und solche mit Festpreis wo das beschränkt wird.

Zu den Versionen, die ihr den Kunden vorstellt: sind das immer vollständig funktionsfähige Systeme, die theoretisch so live gehen könnten?

SW: Es gibt auch Fälle, wo der Kunde irgendwann in den Wunsch hat und wir übergibt uns, wie man das machen könnte, machen in den Entwurf und stellen den dem Kunden den Entwurf im Testsystem vor. Dann kann der Kunde testeten und sagen „Ja, eigentlich wäre uns lieber, wenn das hier ein bisschen anders funktionieren würde“.

FS: Also dass der Kunde einbezogen wird und Feedback geben kann. Macht ihr das häufig?

SW: Wenn es sich um ein Projekt handelt, das langfristig von uns betreut wird (und entsprechend nach Aufwand abgerechnet wird), dann passiert das öfters. Aber eigentlich haben wir mehr Projekte mit begründeten Anforderungen.

Es gibt immer auch noch die Möglichkeit, dass man sagt „OK, im Vertrag steht, der Kunde hat folgende Anforderungen, das wird im Rahmen des Projekts umgesetzt.“ Wenn zusätzlich Wunsch kommt, ist das auch möglich, die werden dann aber nach Aufwand abgerechnet.“ Also dass man in der Sache abstimmt, dass bestimmte Sachen im Projekt gemacht werden müssen, dass wir uns dazu verpflichten. Andererseits darf der Kunde Änderungswünsche haben, aber das ist dann etwas zusätzliches. Ihm muss klar sein, dass wir zusätzlich Zeit brauchen und es nach Aufwand berechnet wird.

FS: Ihr habt also mehrere Sorten von Projekten, zusammengefasst: einmal erster Zeitplan, festes Budget, dann komplett nach Aufwand bezahlt, Kunde mit einbezogen, Änderungswünsche, dann gemischt mit festem Budget aber Änderungswünsche sind erlaubt, kosten je doch extra. Und dann noch so Wartungsverträge, wo ihr in den Shops langfristig betreut und auch nach Zeit abrechnet.

SW: Wenn wir so in den Shops betreiben, ist das meist mit in mehreren Projekten verbunden, das dann in den Betriebsverträgen übergriffen wird.

FS: Was würdest du sagen, welche Art von Projekten am besten funktionieren?

SW: Das was man nach Aufwand abrechnet. Weil es da einfacher ist, die Aufgaben zu verteilen und dem Kunden in gewissen Abständen die Ergebnisse zu zeigen. Er entscheidet, ob es online geht.

FS: Dass der Kunde also mehr einbezogen wird?

SW: Ja, und es kommt nicht so viel Zeitdruck für den Programmierer. Also es gibt schon Zeitdruck, man schätzt die Aufgabe ja. Aber das ist der Zeitdruck, den man sich selber aufregelt, nicht vom Kunden aufregelt bekommen. Vom unternehmerischen her ist es auch sicher. Jedem Monat kann man dem Kunden die Aufgaben in Rechnung stellen und muss damit nicht bis zum Ende des Projekts warten.

FS: Arbeit in den Programmierern denn besser, wenn sie die zusätzlichen Zeitdruck nicht haben?

SW: Das ist eigentlich unterschiedlich, manche Leute arbeiten effizienter, wenn sie unter Zeitdruck sind. Aber in der Sache ist die Effektivität der Mitarbeiter, die anders ist die Atmosphäre im Unternehmen. Es ist schon so, dass wenn man Zeitdruck hat, man effizienter arbeitet, aber wenn das zu lang so läuft, sind die Mitarbeiter gestresst und nicht mehr zufrieden, und unzufriedene Mitarbeiter arbeiten auch nicht mehr so konzentriert. Langfristig ist es also sicher besser, wenn der Druck zu haben.

FS: Zeitdruck und finanzielle Sicherheit sowie mehr Einbeziehung vom Kunden sind die wichtigsten Punkte, gibt es sonst noch Unterschiede?

SW: Ja, das stimmt, dass der Kunde mehr einbezogen wird. Wenn wir in der Dadein haben, macht man das gerne öfter Annahmen, dass etwas aufbrennt. Was umgesetzt wird, den wenn wir den Kunden immer wieder fragen müssen, können wir in der Dadein nicht halten.

FS: Versucht ihr dahin zu gehen, die Abrechnung nach Zeit statt mit festem Budget möglichst immer zu machen?

SW: Die meisten neuen Aufträge sind damit verbunden, dass der Kunde in Projekt umsetzen möchte, das kann sich nachher auf längere Zusammenarbeit verlängern. Er benötigt zunächst in funktionierender Version, mit der er zufrieden ist aber er überlegt sich immer z.B. aus Marketinggründen oder Funktionalitätsgründen etwas zusätzliches. Und das wird dann meistens nach Aufwand berechnet.

FS: Ihr habt also meistens Kunden die erst mal in den Shop zum Festpreis möchte, und das macht ihr auch aber sie geht oft in langfristige Zusammenarbeit über.

SW: Ja, das kann so passieren. Es gibt auch Kunden, die sind mit ihrer Endversion zufrieden und dann wird es abgeschlossen. Es kann aber sein, dass er nach ein oder zwei Jahren kommt und Änderungen möchte.

FS: Ihr hattet zumindest schon Projekte, in denen ihr von Anfang an die so typisch agile Modellgefahren habt. Aber das war eher die Ausnahme?

SW: Ja. Es gibt mehr Projekte, die zumindest am Anfang etwas fest abgestimmt haben. Zumindest dass es Anforderungen gibt, für die zum Beispiel zwischen drei und fünf Tagen abgerechnet werden können. Es gibt also in den Rahmen „von bis“, in Grenzen, wie viel maximal abgerechnet wird, auch wenn es länger dauert.

FS: Kommt es häufig vor, dass in diesen Projekten in den mittleren und Anforderungen und Wünschen kommen, die ihr zusätzlich berechnen müsst?

SW: Ja.

FS: Warum ist das so?

SW: Ich habe den Eindruck, der Kunde hat in der Vorstellung vom Projekt, und wenn ihm dann Teilgezeigt werden (das passiert in dem meisten Projekten, damit ersieht was wir schon gemacht haben), dann merkt er, dass er an irgendwo in Funktionalität vorher nicht gedacht hat, und dann kommen die Ideen. Das sind natürlich immer zusätzlich Anforderungen.

FS: Sind das eher zusätzlich Anforderungen an die Erwartung an den in ihr schon arbeitend oder komplett neue Dinge?

SW: Unterschiedlich, das kommt beides vor. Aber wenn es in ganz neue Sache ist, die zum Beispiel nicht ganz damit verbunden ist, was gerade umgesetzt ist, wird abgesprochen, dass damit erst angefangen wird, wenn wir mit den Sachen fertig sind, die wir gerade machen. Ansonsten wird es zwar zusätzlich abgerechnet und geklärt dass es in paar Tagen länger dauert aber wird in die aktuelle Entwicklung eingeschoben.

FS: Zusammengefasst, wo sieht du im gesamten Entwicklungsprozess die größten Probleme, und wo könnt man am besten etwas verbessern?

SW: Beim Testen. Was in der letzten Zeit als Anforderung an die Programmierer steht, ist dass sie die Sache erst selbst testen, damit die Aufgaben nicht so oft hin und zurück gehen, damit sie auch selbst auf die Qualität mehr achten.

FS: Bezieht sich das auch auf die Code-Qualität, dass also bei den Reviews nicht so viel zurückgeht oder nur auf das funktionale Testen?

SW: Bids. Dann die Zitschätzung, damit man auch intern besser weiß wie lang so ein Projekt dauert. Ohne diese Information kann man auch innerhalb des Unternehmens nicht so gut planen. Wenn man ein Projekt zum Beispiel bis Ende des Monats fertig stellen soll, aber weiß, man schafft es Mitte des Monats, weiß man wann man mit dem nächsten anfangen kann. Ohne genaue Schätzung kann man nicht so viele Sachen einplanen, damit nachher nicht zu viel auf einmal gemacht werden muss.

A.3.3 Paraphrasiertes Interview mit Überschriften

Projektmanagement

Thema: Phasen

Frage: Ihr versucht, XP anzuwenden, wie sind da die Erfahrungen?

Was wir von XP ins tzn ist die Aufteilung der Aufgaben im Projekt. Neben dem Online-Tracking-System haben wir dazu in Tafel im Büro. Für jeden Mitarbeiter gibt es in Ziel und die Aufgaben wandern von „open“ über „todo this week“, „in progress“, „testing“, „review“ bis „done“.

Thema: Team Meeting

(Fortsetzung)

Vor dem wöchentlichen Team Meeting werden die Ziele mit den Aufgaben mit aktualisierten Zielschätzungen und ausgedruckt und am Board ausgetauscht. Darunter können berits neue Aufgaben sein, über die sich die Programmierer im Voraus Gedanken machen sollen, um in der nächsten Schätzung abzugeben. Im Team Meeting berichten sie, was sie geschafft haben, wo sie Probleme haben und warum etwas nicht in der vorgeschlagenen Zielschätzung wurde. Dann werden neue Aufgaben zugewiesen und sie können technisch Fragen dazu an Nikolai [Krambrock, Geschäftsführer] gestellt werden.

Frage: Bei Problemen mit den aktuellen Aufgaben: Wird beim Team Meeting versucht, mögliche Lösungen zu finden?

Ja. Wenn ein Problem gibt, wird das abgeräumt und diskutiert im Team untereinander abgestimmt. Bei Unklarheiten in den Aufgaben kommt die Frage an mich. Im Team Meeting kommen dann die Fragen, die klar beantwortet werden können an Nikolai.

Damit bei weiteren Unklarheiten in den Aufgaben bis zum nächsten Team Meeting in der Woche wartet, gibt es in den nächsten zwei Tagen noch in kürzeren Meetings für Rückfragen.

Thema: Deadlines, Abrechnung

Frage: Das Team ist sehr klein, würden bestimmte Sachen in einem größeren Unternehmen besser funktionieren?

Ich glaube, ja. Wenn in Projekten in der bestmöglichen Zielabgeschlossenheit sein muss und viel Sachen länger brauchen, kommt irgendwann zum Zeitdruck. Mit mehr Leuten könnte man mehr Aufgaben gleichzeitig verteilten und die Chance ist größer, schon vor dem Abgabetermin fertig zu sein. Dann gibt es weniger Stress in der letzten Phase.

Frage: Apropos Zeitdruck und Deadlines: Muss jedes Projekt zu einem bestimmten Zeitpunkt fertig sein?

Nein, nicht jedes. Aber bei Projekten, wo noch kein Online-Shop vorhanden ist, möchte der Kunde ihn zu einem bestimmten Zeitpunkt online haben. Wenn der Shop dann noch nicht fertig ist, kann er natürlich nicht online gehen aber wir versuchen, dies Terminezuhalten.

Dieses Projekt mit Deadlines haben klare Anforderungen und werden pauschal abgerechnet. Es kann auch vertraglich festgelegt werden, dass zusätzlich Wünsche und Änderungswünsche

möglich sind. Dies würde dann aber nach Aufwand abgerechnet und dem Kunden muss klar sein, dass wir zusätzlich Zeit benötigen.

Es gibt auch Fälle, wo wir zunächst in den Entwurf machen, dann wird dem Kunden im Teilsystem vorgestellt. Dann kann der Status und direkt Feedback geben. Das passiert öfters bei Projekten, die langfristig von uns betreut und nach Aufwand abgerechnet werden.

Vollständig agile Projekte sind eher die Ausnahme. Es gibt mehr Projekte, die zumindest am Anfang fest abgestimmte Anforderungen haben, für die zumindest ein Zeitrahmen vorgegeben ist. Zum Beispiel dass für ein Anforderung zwischen drei und fünf Tagen abgerechnet werden können, man also ein Grenzwert hat, wie viel maximal abgerechnet wird, auch wenn es länger dauert.

Es gibt auch Projekte, bei denen es sich um Retail Shops handelt, da ist das nicht so streng zeitlich begrenzt, es gibt nur eine grobe Schätzung (z.B. „innerhalb von einem Monat“) je nach der aktuellen Kapazität.

In Budgetverträgen wird ebenfalls nach Aufwand abgerechnet, die sind meistens mit dem ersten Projekt verbunden, das anschließend in den Budgetvertrag übergeht. Bei diesen langfristigen Zusammenarbeiten können dann immer wieder zusätzliche Wünsche kommen.

Und wir haben Projekte, bei denen wir zuerst in Version fertig machen, sind dem Kunden vorgestellt und dann gibt es noch Änderungswünsche.

Frage: Was würdest du sagen, welche Art von Projekt am besten funktioniert?

Das mit Abrechnung nach Aufwand. Das ist einfach, die Aufgaben zu verteilen und dem Kunden in gewissen Abständen die Ergebnisse zu zeigen. Er entscheidet dann, wann es online geht. Es gibt nicht so viel vom Kunden aufregten Zeitdruck für den Programmierer – nur das selbst aufregend durch die eigenen Schätzungen. Vom Unternehmen her ist es auch sicher, da man dem Kunden je den Monat die Arbeit in Rechnung stellen kann und nicht bis zum Ende des Projekts warten muss.

Frage: Arbeiten die Programmierer denn besser, wenn sie diesen zusätzlichen Zeitdruck nicht haben?

Das ist unterschiedlich, manche Leute arbeiten effizienter, wenn sie unter Zeitdruck stehen. Wenn das aber lang so läuft, sind die Mitarbeiter gestresst und unzufrieden und arbeiten dann nicht mehr so konzentriert. Langfristig ist es also sicher besser, wenn sie keinen Druck zu haben.

Frage: Was sind außer Zeitdruck und finanzieller Sicherheit die Unterschiede?

Der Kunde wird mehr involviert. Wenn wir in Deutschland haben, machen wir das oft in Annahmen, dann wenn wir den Kunden immer wieder fragen müssen, können wir die Details nicht inhaltlich.

Anforderungsanalyse/Spezifikation

Thema: Sonderwünsche

Frage: Kommt es häufig vor, dass mittendrin neue Anforderungen und Wünsche kommen, die ihr zusätzlich berechnen müsst?

Ja.

Frage: Warum ist das so?

Ich hab' den Eindruck, wenn dem Kunden Teil gegeben wird (das passiert in dem Moment ist ein Projekt, damit sicher ist, was wir schon gemacht haben), dann merke ich, dass er an irgendwas in Funktionalität vorher nicht gedacht hat und dann kommen die Ideen. Es kommt vor, dass die Änderungswünsche sind oder auch komplette neue Dinge. In dem Fall dass es nichts damit zu tun hat, was gerade umgesetzt wird, wird abgesprochen, dass wir erst damit anfangen, wenn die aktuellen Aufgaben fertig sind. Ansonsten wird es zwar zusätzlich abgeklärt aber in der aktuellen Entwicklung eingeschoben, die dann in paar Tagen länger dauert.

Thema: Entwicklerperspektive

Frage: Bist du mit in die Anforderungsanalyse involviert?

Bis jetzt nicht, die Aufgaben kommen fertig bei mir an und ich bin dafür verantwortlich, sie für die Entwickler zu übersetzen [Anm.: nicht-duitschsprachig sein]. Dazu muss ich verstehen, worum es geht und fähig sein, die Sachen zu erklären.

Frage: Es geht also nicht nur darum, sie von Deutsch nach Englisch zu übersetzen sondern sozusagen auch von Kundensprache nach Entwicklersprache?

In „Entwicklersprache“ übersetzen kann ich sicher natürlich nicht, aber den Entwicklern hilft oft die Perspektive des Kunden, so dass ich schon mal erklären muss, dass etwas so und so aussieht, damit die Funktionalität logisch ist. Sie soll so einfach wie möglich umgesetzt werden aber ganz wichtig ist, dass sie logisch bleibt.

Die Entwickler haben manchmal nur die technische Perspektive und setzen zwar das um, was in den Aufgaben beschrieben worden ist, machen sich dabei aber kein Gedanken, was für einen einfachen Benutzer unpraktisch ist.

Oft kommen Fragen, zu denen es eigentlich nur eine einzige Lösung gibt, wenn man ein bisschen darüber nachdenkt, weil die andere Lösung unpraktisch ist oder nicht ganz logisch.

Ich erkläre die Aufgaben und die Logik dahinter also genau, auch Dinge die aus Kundensicht vielleicht selbstverständlich sind.

Thema: Unklare Anforderungen

(Fortsetzung)

Oft hat der Kunde auch nicht ganz technisches Wissen, weshalb manche Aufgaben nicht eindeutig sind, man könnte sie auf verschiedene Weisen umsetzen.

Frage: Kann es auch sein, dass eine Aufgabe auf mehrere Weisen verstanden werden kann, unabhängig von der technischen Umsetzung? Wie geht ihr damit um, wenn Aufgaben unterschiedlich interpretiert werden können?

Zuerst stimmt der Programmierer mit mir ab und ich überlege, was die logische Lösung ist. Gibt es dann immer noch mehrere Möglichkeiten wird es mit Nikolai abgestimmt, der den direkten Kontakt zum Kunden hat. Vielleicht wurde es im Gespräch klar, aber nicht detailliert beschrieben, oder merkt die Denkweise des Kunden besser. Das Problem ist, dass es so immer etwas länger dauert, bis die Entscheidung getroffen wurde.

Dinge, die so unklar sind, dass beim Kunden zurückgefragt werden muss, werden erst gesammelt, damit es nicht zu vielen Rückfragen gibt.

Thema: Aufwandschätzung

Frage: Wie läuft die Aufwandschätzung ab?

Wenn die Programmierer Aufgaben bekommen, sollen sie selbst schätzen, wie viel Stunden oder Tag sie brauchen. Sie sollen sich dabei überlegen, was zu tun ist, und wie man sie vielleicht umsetzen könnte. Aber ich habe den Eindruck, dass Überlegungen passieren nicht immer so ganz genau, sondern erst, wenn sie die Aufgabe bereits auf dem Board haben und mit der Umsetzung anfangen. Die Schätzung passiert manchmal ziemlich schnell, deshalb denke ich mir, dass können sie noch gar keine Lösung wissen, vielleicht ist das nicht so genau.

Frage: Ist dieses schnelle Schätzen also ein Problem für die Genauigkeit?

Das liegt vielleicht auch daran, dass sie mehrere Aufgaben auf einmal zugewiesen bekommen, die alle geschätzt werden sollen, von denen aber nur einige auf „Todo this week“ gesetzt werden.

Die angefangenen Aufgaben, die nicht innerhalb der vorgesehenen Zeit zu Ende gebracht werden konnten, weil irgendwelche Probleme aufgetreten sind, werden umgeschätzt. Das ist so ein bisschen das Problem, dass sich die Schätzung in der Zwischenzeit ändert.

Frage: Wann wird das denn genau korrigiert?

Es ist oft so, dass ein Programmierer für eine Aufgabe ursprünglich zwei Tage geschätzt hat und wenn er mit der Aufgabe angefangen hat, merkt er, dass er in ein Tag zusätzlich braucht. Die Schätzung wird vor der nächsten Iteration, also vor dem Team Meeting korrigiert, wenn die Aufgaben am Board diskutiert werden und neu zugewiesen werden.

Frage: Würdest du sagen, dass die zweite korrigierte Schätzung genauer ist, oder wird da auch noch oft nachkorrigiert?

Unterschiedlich. Ganz komplizierte Aufgaben werden auch schon mal zwei oder drei mal neu geschätzt. Das ist natürlich aus der Management-Perspektive nicht gut, weil man nicht klar sagen kann, wann wir mit dem Projekt überhaupt fertig sind.

Dazu kommt, dass die Schätzung beim Team Meeting immer noch einmal hinterfragt werden, und Nikolai die Programmierer daran erinnert, dass Nachbesserungen nach Test und Review berücksichtigt werden müssen. Die Programmierer merken dann, dass sie sich mit ihrer geschätzten Zeit doch nicht so sicher sind. Das Problem ist, dass diese Frage kommt eigentlich jedes Wochenende und so stellt man schon beim Team Meeting fest, dass die Schätzung nicht so genau sind.

Entwurf

Thema: Trennung von Entwurf und Programmierung

Frage: Wäre es hilfreich, wenn der Entwurf zuerst gemacht und dabei geschätzt würde?

Das Problem dabei könnte sein, dass der Programmierer nicht so in gute Gefühle hat, wenn er Entwurf macht und wenn er anfängt zu programmieren. Das vermischen ist nicht, deshalb macht er bei der Schätzung auch noch keinen Entwurf und deswegen weiß er nicht so genau,

wi vil Zeit er benötigt. Wenn die Leute für jede Schätzung erst mal in den Entwurf machen würden, dann dauert die Schätzung selbst auch viel länger.

Thema: Falscher Entwurf

Frage: Wenn ein Programmierer sich nun einen Entwurf macht und Vor- und Nachteile verschiedener Lösungen abwägt, aber selbst noch nicht so viel Erfahrung hat, kann er das vielleicht noch gar nicht so gut einschätzen. Kommt es oft vor, dass etwas als gut beurteilt wird aber es dann doch irgendwann nicht weiter geht und alles anders gemacht werden muss?

Ja, solche Fälle hatten wir schon.

Frage: Ist das eher ein Anfänger-Problem oder hat es auch mit Magento zu tun?

Der Programmierer erzählt schon, dass Magento nicht sehr übersichtlich ist und unklar ist, was passieren kann, wenn man irgendwo etwas ändert. Magento selbst kann also in Teil des Problems sein, aber wenn der Programmierer ein bisschen Erfahrung hat, kennt er die Vorgeschichte und weiß auch was problematisch sein kann. Die Erfahrung macht da, glaube ich, viel aus.

Thema: Finden von Lösungen

Frage: Hast du ein bisschen Einblick in die Entwurfs-Aktivitäten?

Nicht immer. Manchmal gibt es mehr technische Lösungen, dann überlegt sich der Entwickler, wie er sie angehen will und wir diskutieren die Unterschiede und die Vor- und Nachteile. Wenn ich den Eindruck habe, dass in der Lösung etwas Besseres ist, kann ich da auch ein bisschen Anweisungen geben. Aber das ist nicht oft der Fall, da die Sachen mehr technische Natur sind, das stimmen die Entwickler eher mit Nikolai ab.

Frage: Liegt es denn in der Entscheidung der Entwickler?

Es ist schon das Ziel, dass die Entwickler mehr Sachkenntnis und selber in Angabe machen, was die beste Lösung ist.

Der gewünschte Vorgeschichte ist, dass die Entwickler in den Team Meetings in Problemdarstellung und fröhlich Vorschläge wie man sie lösen kann und dann wird darüber diskutiert und der beste Entwurf ausgewählt. Manchmal sehen sie schon bei der Überlegung, dass in Lösung gar nicht geht und nur noch in übrig bleibt. Dann kann er selbst entscheiden, dass es so geht und im Team Meeting wird einfach bestätigt, was er gerade macht.

Frage: Also hilft diese Art von Entwurf schon oft, herauszufinden, was funktioniert und was nicht, bevor man anfängt, etwas zu programmieren, was vielleicht gar nicht funktionieren kann.

Ja. Ich bin mir allerdings nicht sicher, ob das wirklich in jedem Fall passiert [siehe auch: Aufwandschätzung].

Programmierung

Thema: Pair Programming

Frage: Noch einmal zur Absprache untereinander: Wird auch Pair Programming eingesetzt?

Es arbeitet man ist nicht unbedingt, nur die die wenigere Erfahrung haben, arbeiten anfangs mit jemand anderem zusammen. Oder die Programmierer setzen sich zusammen, wenn es ein Problem gibt, und suchen gemeinsam eine Lösung, aber es ist nicht so, dass jede Aufgabe gemeinsam gemacht wird. Also nur wenn jemand ganz neu ist oder nicht alleine weiterkommt.

Frage: Geht es so schneller?

Was schneller geht, bin ich nicht sicher, wir haben keine Erfahrungswerte zu konsolidiertem Pair Programming. Das kann auch daran liegen, dass wir nicht so viele Mitarbeiter haben und dann ist es vielleicht doch schneller, wenn sie gleichzeitig Aufgaben machen. Aber wenn sie zwei bei einem Problem zusammen setzen, sind sie schon schneller damit, eine Lösung zu finden, habe ich den Eindruck.

Frage: Du kannst also nicht konkret sagen, warum ihr euch entschieden habt, das nicht durchgängig zu machen? Würde das Vier-Augen-Prinzip nicht helfen, mehr Fehler im Voraus zu finden?

Schwierig zu sagen, da wir es nicht ausgetestet haben. Es kann gut sein, dass wenn man auf etwas nicht achtet, dass dem anderen gar nicht auffällt. Es fehlt nur bisher die Möglichkeit, das auszuprobieren.

Test

Thema: Iterationen zwischen Implementierung und Test

Frage: Gibt es denn das Problem, dass die Programmierer Fehler übersehen?

Oft, gerade bei Mitarbeitern mit weniger Erfahrung, bei neuen Aufgaben lang, bis sie fertig sind, weil in der Test-Phase Fehler gefunden werden und in Aufgaben manchmal vom Test zurück nach „in progress“ geschoben wird.

Es dauert natürlich etwas, bis der Test der Test durchführt, ich mache das mindestens einmal täglich. Während die Aufgabe in „testing“ hängt, beschäftigt sich der Programmierer dann schon mit einer neuen Aufgabe und hat schließlich mehr angefangen Aufgaben gleichzeitig, die die wenigere schneller zu Ende kommen.

Thema: Test während der Entwicklung

Frage: Gibt es keine Tests, die die Programmierer selber durchführen?

Doch, sie sollen ihre Sachen zuerst selbst testen. Aber sie haben dabei in andere Perspektiven und finden Sachen nicht, die in normaler Benutzung nicht.

Frage: Werden während der Entwicklung Test-Tools wie Selenium eingesetzt?

Nicht während der Entwicklung.

Frage: Kann man also sagen, dass das Testen während der Entwicklung noch ein Problemfeld ist? Dass die Programmierer mit ihrer eingeschränkten Perspektive und unsystematischem Testen Fehler nicht finden?

Ja.

Frage: Gab es mal Versuche, während der Entwicklung systematischer zu testen?

Es wird versucht, mehr Wert darauf zu legen, dass die Programme rüstet sind und dafür auch mehr Zeit investiert. Es ist in Projekten, wenn es im Testbereich in den ersten drei, vier Minuten in Fehler gefunden wird und sich allseits zitiert.

Thema: Manuelle Tests

Frage: Wie ist das mit den Tests, die du manuell machst. Probiert du alle Anwendungsfälle durch, hast du da ein System?

Bei Erweiterungen, die dem System in bestimmten Fähigkeiten hinzufügen, orientiere ich mich natürlich daran, was geändert wurde. Abhängig davon, an welcher Stelle im Shop das ist, überlege ich mir, was kaputt gehen könnte. Dann gibt es in paar Sachen die man standardmäßig durchgeht, die wichtigsten Funktionen in einem Online-Shop sind Such und Checkout. Wenn sich z.B. im Checkout etwas geändert hat, wird der zu erst manuell getestet und wenn alles funktioniert, kann in Selenium-Test daraus erstellt werden.

Frage: Testest du mit verschiedenen Daten verschiedene Sonderfälle oder gehst du nur Standard-Fälle durch?

Ich versuche schon, verschiedene Fälle zu testen, beim Checkout zum Beispiel immer die Fälle „Eingeloggt“, „Gast“ und „Kunde“, der sich gerade anmeldet“. Oft funktionieren etwas beim Gast, beim eingeloggt Kunden aber nicht mehr.

Manche Schritte werden sich dabei natürlich abnormale sind es nur zwei, drei Fälle. In solchen Fällen muss ich mehr präventiv Fälle für irgendwelche Systemeigenschaft testen. Ich hatte zum Beispiel mal in einem Fall, wo es um Steuerberechnung ging, die zusätzlich von der Zahlungsart abhängig war. Da musste man alle Kombinationen durchgehen und die habe ich dann in einer Excel-Tabelle dokumentiert.

Thema: Seiteneffekte

Frage: Kommt es vor, dass Sonderfälle übersehen werden, die dann später zu einem Fehler führen?

Manchmal. Es kommt vor, dass ich keinen Fehler finde und der Code auch das richtige Verhalten zeigt. Das Problem mit Magento ist, dass oft irgendwelche ganz unerwartete Fehler auftauchen. Man hat etwas programmiert und damit in einer Funktion kaputt gemacht, ohne damit zu rechnen.

Thema: Regressionstest, Probleme mit Selenium

(Fortsetzung)

Dazu gibt es die Selenium-Tests: Die werden entwickelt, wenn die Aufgabe schon fast durch ist. Sie werden als Überwachungssystem genutzt, um zu überprüfen ob spätere Änderungen bei bereits fertiggestellten Funktionen kaputt gemacht haben.

Frage: Können die Selenium-Tests denn alles abdecken?

Was man mit Selenium zum Beispiel nicht testen kann, ist ob das System Emails verschiebt. Das Testen von einfachen Sachen, die man immer wieder macht, wird aber gut abgedeckt, das finde ich gut.

Was für mich als nicht-Programmierer problematisch ist, ist dass die Tests manchmal nicht sofort funktionieren und man von Hand noch Kommandos eingeben muss.

Frage: Manuelle Tests sind also zunächst weniger aufwändig?

Mit einem Selenium-Testverbringen ich manchmal mehr Zeit als mit einem manuellen Test. Aber bei Tests, die immer wieder ausgeführt werden müssen, und das sind natürlich alle, ist es immer besser, sie von einem automatisierten Test machen zu lassen.

Frage: Werden es denn irgendwann zu viele, da sie ja doch recht lange brauchen, um durchzulaufen?

Man versucht, kein Tests zu wiederholen, die genau das gleiche machen, zum Beispiel die selbe Funktion mit verschiedenen Produkten. Das macht nur Sinn wenn es notwendig ist. Unterschiedliche Tests, zum Beispiel bei einfachen und konfigurierbaren Produkten. Auf der anderen Seite, versucht man, pro Test nicht zu viele Eigenschaften zu testen, damit wir bei einem Fehlschlag schnell referenzieren, was genau das Problem ist.

Wie lang ein Test-Durchlauf ist, hängt von der Anzahl der Tests ab, von der Dauer der einzelnen Tests und davon, ob Fehler gefunden werden. Wenn Fehler gefunden werden, kann es zum Beispiel in einem Timeout von 3000 Sekunden [sic] geben, dann dauert es natürlich länger. Die Regel ist bei über 40 Test Cases zwischen 5 und 8 Minuten.

Frage: Ist diese Dauer ein Problem?

Ja, wenn mehr der Leute gleichzeitig an einer Sache arbeiten. Wenn in Programmierer Änderungen hochspielen, werden die Selenium-Tests automatisch ausgeführt. Wenn ich gleichzeitig in der Instanz etwas manuell teste, schlagen immer in paar der automatisierten Tests fehl, die durch mein manuelles Testen gestört werden. Ich muss auf die seriell in der Instanz testen, weil ich mir da sicher sein kann, dass es die alleraktuellste Version ist.

Integration

Thema: Testsystem

Frage: Die Entwickler entwickeln und testen zunächst auf ihren Instanzen ihre eigenen Änderungen und auf dem Test-System kommen alle aktuellen Änderungen zusammen, richtig?

Nicht ganz. Es ist nicht so dass jeder auf seiner lokalen Magento-Instanz programmiert, bevor man etwas neu entwickelt, lädt man sich die aktuelle Version herunter. Wenn mehr der Leute gleichzeitig an einem Projekt arbeiten, kann es passieren, dass in Programmierer etwas schnell zu Ende bringt und es in einem Konflikt gibt, wenn der nächste seine Änderungen hochspielt.

Thema: Integration von Templates und Modulen

(Fortsetzung)

Dazu kommt, dass wir die Templates nicht selbst entwickeln, das macht immer jemand extern und oft geht da die Kommunikation über den Kunden. Das Zusammenspielen von Modulen und Templates ist auch ein Integrationsproblem. Es kann sein dass in Templates etwas kaputt macht oder nicht berücksichtigt.

Es ist auch problematisch, wenn die Programmierer etwas machen, was vom Template abhängig ist. Das zu vermeiden, ist in klarer Anforderung bei uns. Die Module müssen selbständig funktionieren, wenn man das vergisst, wird es problematisch.

Wenn man auf die Unabhängigkeit vom Template achtet und dann etwas nicht funktioniert, kann man erstmal das Template abschalten und schauen, ob es ohne das Template funktioniert. Falls ja, ist es wahrscheinlich ein Problem im Template.

Thema: Integration von Modulen

Frage: Wie ist es mit Modulen untereinander, gibt es da Konflikte?

Ja. Wir achten darauf, dass Module, die wir neu programmieren, mit den bereits integrierten Sachen zusammen funktionieren. Es kann aber zum Beispiel passieren, dass wir nachträglich in Zahlungsmodul installieren und es dann kracht, weil es die gleichen Klassen überschreibt, wie in anderen. In dem Fall müssen wir nachträglich das gleiche Modul anpassen.

Frage: Sind Fremde Module ein großes Problem? Nutzt ihr überhaupt viele fremde Module?

Problem gibt es da nicht so oft aber wir nutzen auch nicht viel. Nur bei Sachen die man eigentlich nicht anders machen kann, wie Bonitätsprüfung, Zahlungsanbieter...

Sonstiges

Thema: Verbesserungsbedarf

Frage: Zusammengefasst: Wo siehst du im gesamten Entwicklungsprozess die größten Probleme, und wo könnte man am ehesten etwas verbessern?

Beim Testen. Auch dass die Programmierer selbst mehr tun und mehr auf die Qualität achten.

Dann die Zeitschätzung, damit man intern auch besser weiß, wie lang so ein Projekt dauert. Ohne diese Information kann man auch innerhalb des Unternehmens nicht so gut planen.

Vielleicht liegt das Problem auch an der Erfahrung der Programmierer. Man sieht, dass diejenigen, die noch nicht so lang im Unternehmen sind, länger brauchen um die gleichen Funktionalität umzusetzen und sie auch nicht so gut schätzen können.