



LoQutus – de TU Delft

Resultaten survey rond
Lean Software Development

December 2011

LoQutus
Latemstraat 93
B-9830 Sint-Martens-Latem
Tel +32 9 242 83 00
Fax +32 9 242 83 01
info@loqutus.com
www.LoQutus.com

INHOUDSTAFEL

Inhoudstafel	2
1. Overzicht	3
1.1 <i>Executive summary</i>	3
1.2 <i>Spreiding respondenten</i>	3
1.3 <i>Opzet van het onderzoek</i>	5
1.4 <i>Methodiek</i>	5
2. Deel 1: spreiding van de respondenten	7
3. Deel 2: Evaluatie van heersende gebruiken in software-ontwikkeling	8
3.1 <i>Overzicht van heersende gebruiken</i>	8
3.2 <i>Succesratio</i>	9
3.3 <i>Evaluatie van technieken</i>	10
3.3.1 <i>Documenteren van de architectuur</i>	10
3.3.2 <i>Projectmanagement best practices</i>	10
3.3.3 <i>Gebruik van (code-niveau) tools draagt niet bij</i>	11
3.4 <i>Wat als doelstellingen niet gehaald worden?</i>	11
3.4.1 <i>Redenen</i>	11
3.4.2 <i>Reactie</i>	11
4. Deel 3: Evaluatie van heersende gebruiken in innovatietrajecten	13
4.1 <i>Meest effectieve innoverende technieken</i>	13
4.2 <i>Belemmeringen voor innovatie</i>	14
4.3 <i>Conclusie</i>	14

1. OVERZICHT

1.1 EXECUTIVE SUMMARY

Uit een studie bij 34 organisaties met een software ontwikkelingsdepartement einde 2010 blijkt hoe een divers gamma aan ontwikkelings-methodieken in gebruik is. Zowel watervalmodellen als Agile ontwikkeltechnieken worden door een aanzienlijk deel van de organisaties gebruikt (vaak komen beide “uitersten” binnen dezelfde organisatie voor). Een aantal ontwikkeltechnieken wordt echter wel door een grote meerderheid van de organisaties gebruikt en kunnen dus als *de facto* standaarden worden aanzien.

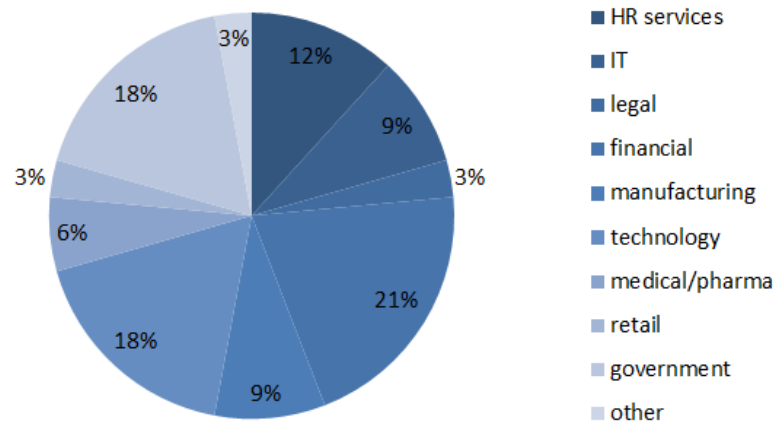
Succesvol projecten opleveren is zeker niet de standaard. Heel wat organisaties hebben moeite om tijd en budget onder controle te houden. Het resultaatcriterium klantentevredenheid daarentegen scoort wel aanzienlijk beter.

Succesvolle organisaties onderscheiden zich voornamelijk door **projectmanagement** (inschatting, risicobeheer, reviews) **en aandacht voor architectuurwerk**. Agile development practices dragen volgens deze rondvraag niet bij tot de slaagkansen van een project.

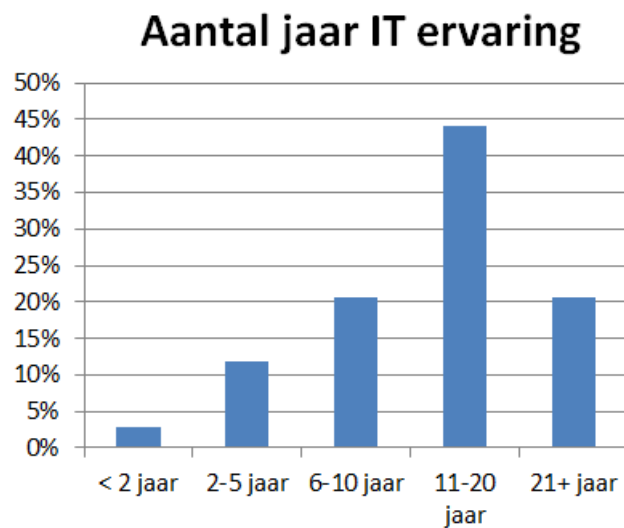
Op het vlak van **innovatie** richten meer succesvolle organisaties zich eerder op trajecten om te verbeteren met betrekking tot het **aspect tijdsbeheer**, terwijl naar de oplevering toe de **klantentevredenheid** blijft primeren. Een hogere maturiteit met betrekking tot innovatie leidt uiteindelijk wel vaker tot het leveren van kwalitatieve, flexibele systemen.

1.2 SPREIDING RESPONDENTEN

De vragenlijst werd breed uitgestuurd, via contacten van zowel LoQutus als de SERG onderzoeksgroep aan de TU Delft én eveneens voorgelegd op een seminarie georganiseerd door LoQutus. Als gevolg daarvan behoren de respondenten tot organisaties die onderling verschillende in karakteristieken, zowel naar sectoren als naar omvang. De sectoren « Financial » en « Government » en « IT-services » zijn het sterkst vertegenwoordigd onder de respondenten. De spreiding naar omvang is evenredig verspreid tussen de categorieën van 1-10 tot en met 1000+. De ontvangen resultaten dekken voornamelijk de gebruiken voor software ontwikkeling van informatiesystemen af, zowel interne applicaties, ERP-systemen als internetsystemen.

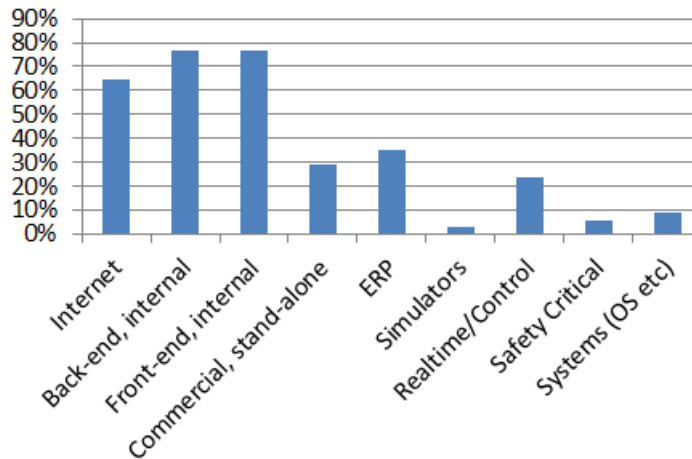


Figuur 1 Deelnemende organisaties per sector



Figuur 2 Aantal jaren ervaring per respondent

Type Projecten



Figuur 3 Types van softwareprojecten die de organisatie van de respondent uitvoert

De vragenlijst is bezorgd aan contactpersonen van de aangesproken bedrijven, met als vraag deze te bezorgen aan personen binnen de organisatie die voldoende zicht hebben op de software ontwikkelingsmethodiek. De grote meerderheid van de respondenten zijn ofwel IT manager, ofwel technical lead/architect. Iets meer dan 60% van de ondervraagden heeft meer dan 10 jaar ervaring in IT.

1.3 OPZET VAN HET ONDERZOEK

Eind 2010 organiseerde LoQutus in samenwerking met de Technische Universiteit Delft een rondvraag over de huidige gebruiken in softwareontwikkeling. Het doel was zowel de huidige gebruiken in kaart te brengen, als na te gaan hoe(veel) organisaties op dit gebied innoveren. Aan de deelnemende organisaties werd naast het in kaart brengen van gebruikte technieken op het gebied van software ontwikkeling ook gevraagd een aantal outputs (budget, tijd, kwaliteit...) van typische projecten in te schalen. Op basis daarvan werd onderzocht welke factoren deze outputs beïnvloeden.

1.4 METHODIEK

Een vragenlijst werd bezorgd aan de aangesproken bedrijven, met als vraag deze te bezorgen aan personen binnen de organisatie die voldoende zicht hebben op de software-ontwikkelingsmethodiek (de respondenten).

De focus van de vragenlijst werd gelegd op gangbare gebruiken. Indien meerdere gebruiken binnen dezelfde organisatie gangbaar waren, werd gevraagd om een recent, representatief project te gebruiken waarbij de respondent zelf betrokken was.

De vragenlijst bestond uit drie delen:

1. In het eerste deel werd de **spreiding van de respondenten** bevestigd: zowel de achtergrond van de respondent als diens organisatie.
2. Het tweede deel onderzocht en evalueerde de **huidige gebruiken binnen de software-afdelingen**.
 - a. Aan de respondenten werd gevraagd om **per ontwikkeldiscipline** aan te geven **welke technieken op dat moment gebruikt werden**. Deze disciplines verdeelden we onder in (i) vereistenbepaling, (ii) analyse & ontwerp, (iii) ontwikkeling, (iv) testen, (v) kwaliteitsbewaking, (vi) planning & opvolging en (vii) inschatting.
 - b. Per discipline werd onderzocht **in welke mate er gebruik gemaakt wordt van softwaretoepassingen** (om de activiteiten in deze discipline te ondersteunen), zoals bijvoorbeeld het gebruik van een versiebeheersysteem voor code in de ontwikkelfase.
 - c. De respondent mocht de **succesratio** aangeven op het respecteren van de aspecten tijd, budget, volledigheid, kwaliteitsniveau, *team spirit* en klantentevredenheid. Daarbij werden vier intervallen gehanteerd, naargelang de regelmaat waaraan het criterium voldaan werd:

Zelden (< 10%)
Soms (10-25%)
Vaak (26-75%)
Meestal (>75%)
 - d. Daarnaast werd ook de **reactie** gevraagd **wanneer de planning niet gehaald wordt**. De impact daarvan op functionaliteit, timing, overwerk en/of toevoeging van medewerkers aan het project werd eveneens in kaart gebracht.
 - e. Meest voorkomende **oorzaken** van het overschrijden van de planning.
3. Het derde deel behandelde de **richting van innovatietrajecten en hun evaluatie**.
 - a. Voor een aantal **opkomende technieken** werd gevraagd **in welke mate hiermee geëxperimenteerd wordt en** of deze technieken als toegevoegde waarde gezien worden en/of eenvoudig aan te leren.
 - b. Wat zijn de **grootste uitdagingen** met betrekking tot innovatie?
 - c. Wat zijn de **evaluatiecriteria om innovatietrajecten te beoordelen?**
 - d. Welke **voordelen** hebben deze trajecten uiteindelijk opgeleverd?

2. DEEL 1: SPREIDING VAN DE RESPONDENTEN

Op basis van de gegevens over de deelnemers kunnen we concluderen dat **de nodige spreiding aanwezig is om over software-ontwikkeling in een brede context uitspraken te kunnen doen**. De respondenten zijn namelijk niet beperkt tot een specifiek domein of doelgroep.

Anderzijds merken we wel op dat we met deze resultatenlijst **geen conclusies** kunnen formuleren **met betrekking tot een specifiek domein of organisatie**. Daarvoor is het aantal meetpunten te beperkt.

3. DEEL 2: EVALUATIE VAN HEERSENDE GEBRUIKEN IN SOFTWARE-ONTWIKKELING

3.1 OVERZICHT VAN HEERSENDE GEBRUIKEN

De beperkte lijst van technieken die door meer dan twee derde van de respondenten wordt aangegeven geeft enerzijds een aantal standaarden aan, en impliceert anderzijds dat er grote variatie bestaat in de ontwikkeltechnieken die binnen de verschillende organisaties gebruikt worden.

- Op het gebied van **ontwikkelmethodiek** is het niet verwonderlijk dat de **'waterval'-ontwikkeltechniek nog steeds sterk verspreid** is. Deze wordt wegens diens gekende voordelen binnen een bepaalde context al een hele tijd gebruikt en is dus ingeburgerd. Verder rapporteerden **47% van de ondervraagden dat Scrum¹ gehanteerd wordt** binnen de organisatie.
 - Op het vlak van **vereistenbeheer** wordt voornamelijk gegrepen naar **use cases en naar tekstverwerkingssoftware**. Use cases komen vanuit de wijdverspreide UML. Deze techniek is breed bruikbaar en wordt onder verschillende vormen gepromoot (Smart use cases, *Agile modeling*, RUP). Vereisten worden verder eerder in de vorm van een document beschreven dan op andere – meer visuele - manieren zoals *whiteboard schetsen* of het gebruik van *requirements tools*.
- **Unit testing**. Deze vroege en automatische vorm van testen is **breed ingeburgerd**. De reden hiervoor ligt volgens ons in de **beperkte leercurve** en de **relatief eenvoudig te berekenen ROI**. Ondanks de opkomst van *automated builds* en continue integratie komt **manuele integratietesting** nog steeds het meest voor.
- Op het gebied van **tooling** ter ondersteuning van het software-ontwikkelingsproces komen **issue tracking en testing** sterk naar voor. Deze hebben een sterke documenterende werking. *Issue tracking* komt voort uit projectgericht werken en service management en is niet noodzakelijk IT-gericht (kijk maar naar *ticketingsystemen*).

¹ Linda Rising, Norman S. Janoff, "The Scrum Software Development Process for Small Teams," IEEE Software, vol. 17, no. 4, pp. 26-32, July/Aug. 2000

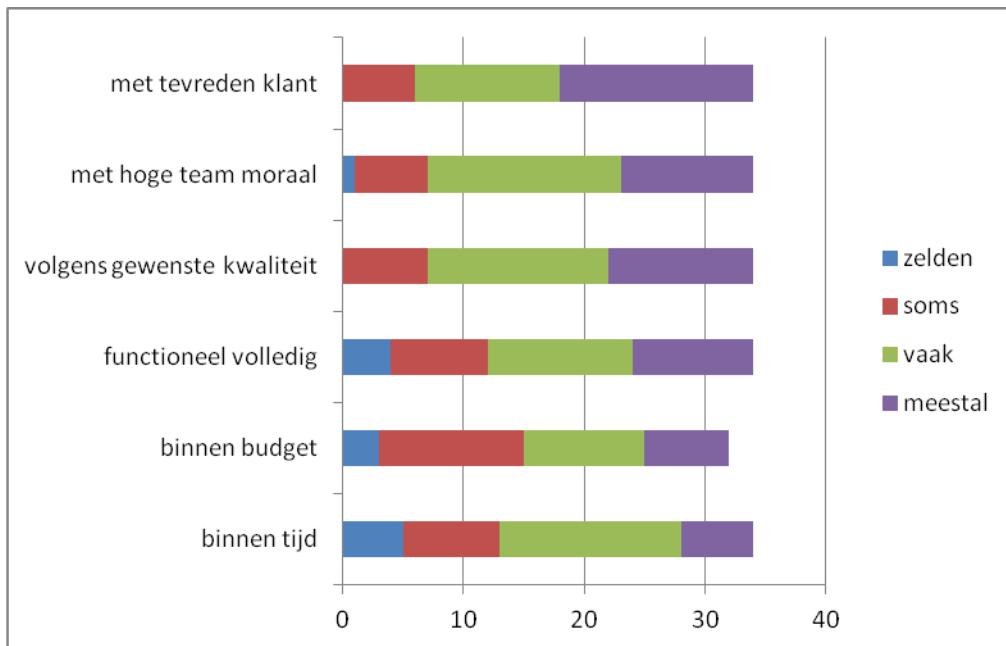
Software Ontwikkelingstechniek	% van organisaties die deze techniek gebruikt
Methodiek - Waterval methode	68%
Requirements – Use Cases	68%
Requirements – Word processor / text editor	68%
Ontwikkeling - Code standaarden	74%
Ontwikkeling – Unit testing	79%
Testen – Unit Testing	91%
Testen – Manual Integration Testing	76%
Planning & Tracking – Planning Meeting	85%
Planning & Tracking – Issue Tracking	74%
Tooling – testing tool	71%

Figuur 4 10 van de 96 opgelijste technieken worden door meer dan twee derde van de deelnemende organisaties gebruikt

3.2 SUCCESRATIO

Uit de resultaten blijkt dat de criteria 'met een tevreden klant' en 'met de gewenste kwaliteit' het meest bereikt worden. Ongeveer de helft van de respondenten rapporteerde dat in meer dan 75% van de projecten de klant tevreden is.

Op andere criteria wordt zwakker gescoord. Vooral de criteria 'binnen budget' en 'binnen tijd' worden niet even vaak gerealiseerd: meer dan een derde van de respondenten geeft aan dat deze criteria in minder dan 25% van de projecten gehaald worden.



Figuur 5 Het bereiken van outputcriteria in termen van een % van projecten die daar in slagen per organisatie, uitgetekend over de gehele populatie

3.3 EVALUATIE VAN TECHNIEKEN

Wanneer we deze resultaten statistisch evalueren, stellen we vast dat **de meer succesvolle organisaties zich maar op een beperkt aantal domeinen onderscheiden**: (i) documenteren van de architectuur, (ii) projectmanagement *best practices*, Automatisering binnen software ontwikkeling (naar testing, builds, versionering) blijkt niet bij te dragen aan een verhoging van één of meerdere outputcriteria.

3.3.1 Documenteren van de architectuur

Het valt op hoe het documenteren van het ontwerp (de architectuur) bijdraagt aan een beter resultaat. De manier waarop varieert (van *whiteboard* over tekentool tot software modeleringsomgeving), maar het is wel duidelijk dat dit aspect bijdraagt tot verbeteringen in alle outputcriteria.

Het feit dat gewezen wordt op het gebruik van teken- en software modelerings *tools* lijkt erop te wijzen dat **een visuele representatie hiervoor de meest aangewezen aanpak** is (in tegenstelling tot onze vaststellingen voor de beschrijving van vereisten).

3.3.2 Projectmanagement best practices

Ten tweede merken we een aantal elementen op van goed projectbeheer: risicobeheer, inschattingen, reviews. Het feit dat deze aspecten hier aanwezig zijn lijkt erop te wijzen dat goed projectbeheer niet gangbaar is in alle deelnemende organisaties. Inschatten (via

analogie) vereist dan weer ervaren teamleden met de nodige achtergrond in vergelijkbare projecten.

3.3.3 Gebruik van (code-niveau) tools draagt niet bij

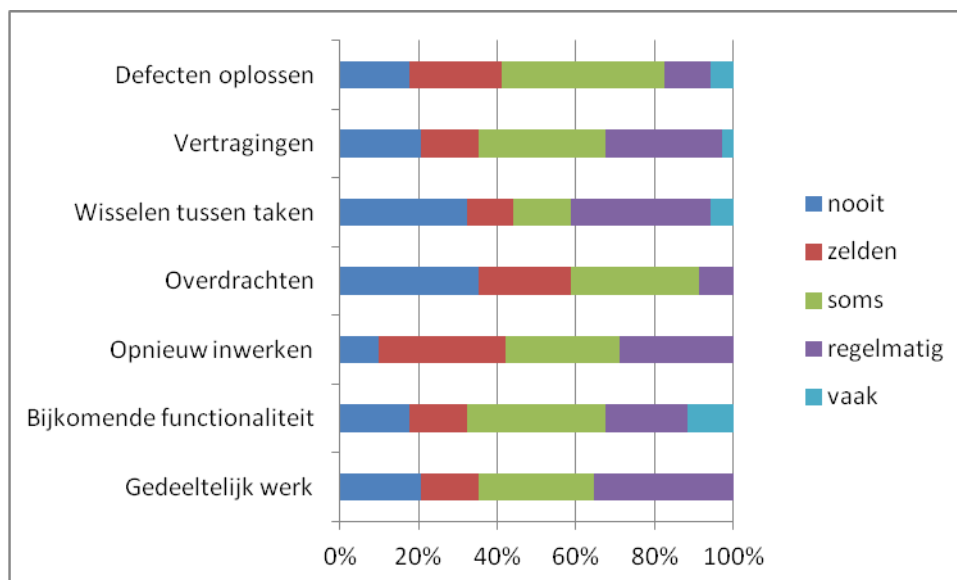
Het gebruik van softwaretools (zoals automatische *builds*, continue integratie, automatische testen en *code refactoring*) ter ondersteuning van het ontwikkelproces is al langer aan de orde. Toch lijken deze elementen niet het verschil te maken voor organisaties met betere resultaten.

3.4 WAT ALS DOELSTELLINGEN NIET GEHAALD WORDEN?

3.4.1 Redenen

De vaakst aangehaalde redenen zijn het wisselen tussen taken (bv. tussen verschillende projecten) en het opleveren van gedeeltelijk werk (werk dat niet meteen afgewerkt wordt).

Het overdragen van werk komt het minst voor: blijkbaar blijven ontwikkelaars op een project eens ze ermee gestart zijn. De reden 'Wisselen tussen taken' geeft de grootste tegenstellingen: sommige organisaties vermijden dit ('zelden' tot 'nooit' voor meer dan 40% van de organisaties), terwijl de andere 40% regelmatig tot vaak op dit scenario overgaat.



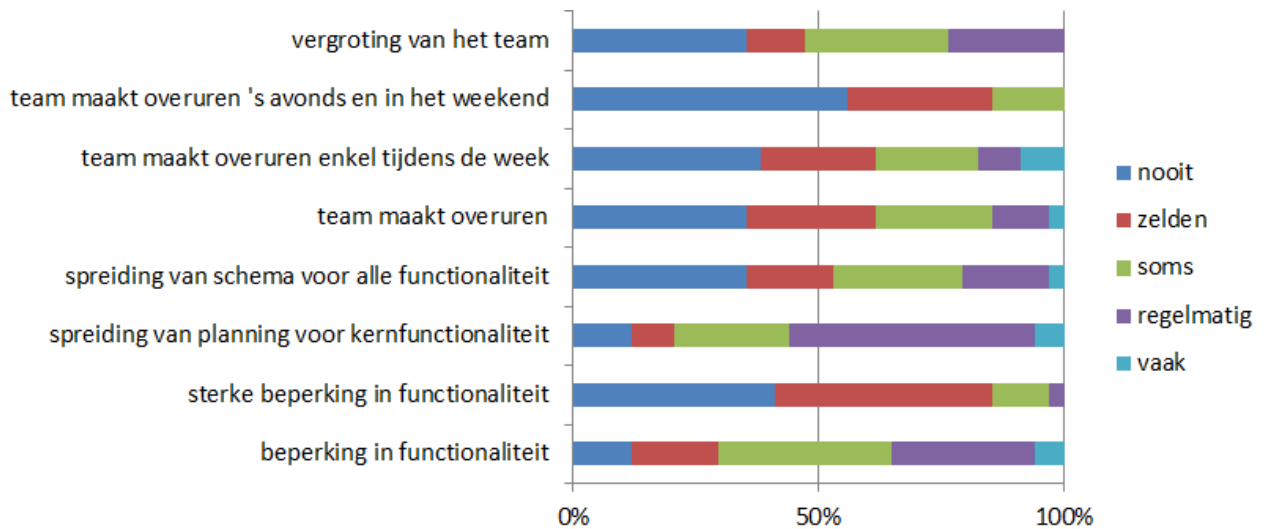
Figuur 6 Waarom doelstellingen niet behaald worden

3.4.2 Reactie

We stelden de volgende vraag aan de respondenten: indien software-implementatieprojecten de vooropgestelde planning niet halen, hoe wordt er dan gereageerd in termen van (i) functionaliteit, (ii) aanpassing van de timing, (iii) overwerk en/of (iv) toevoeging van medewerkers op het project?

Blijkt dat meestal wordt ingespeeld op het bereik (de scope) van de uitvoering: de oplevering van functionaliteiten wordt meer gespreid in de tijd of er wordt functionaliteit geschrapt. **Gezien eerdere rapporten² rond niet-gebruikte functionaliteit in software lijkt dat een verstandige keuze.**

In meer dan 50% van de ondervraagde organisaties wordt er nooit 's avonds of in het weekend overgewerkt. Samen met sterke beperkingen in functionaliteit is dit de minst populaire maatregel.



Figuur 7 Reactie op niet behalen doelstellingen

² Jim Johnson. *The Standish Group International Inc.* 2002.,
Ralph Bohnet, Gerard Meszaros: *Test-Driven Porting. AGILE 2005: 259-266*

4. DEEL 3: EVALUATIE VAN HEERSENDE GEBRUIKEN IN INNOVATIETRAJECTEN

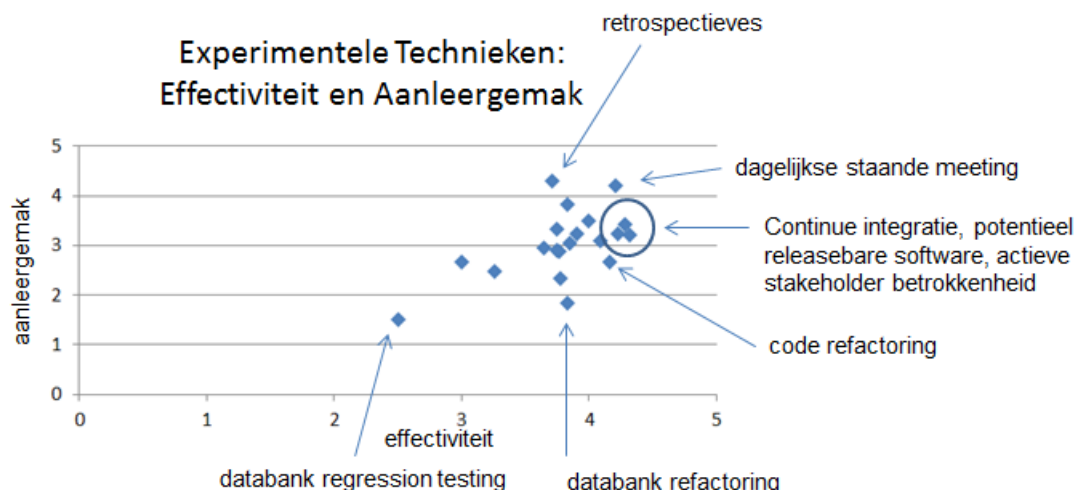
4.1 MEEST EFFECTIEVE INNOVERENDE TECHNIEKEN

Technieken die het meest gebruikt worden, als meest effectief ervaren worden én eenvoudig aan te leren, zijn de volgende:

- continue integratie (*continuous integration*),
- dagelijkse statusmeeting (*daily standup meeting*),
- uitvoerbare specificaties,
- actieve betrokkenheid van stakeholders,
- potentieel releasebare software
- collectief code-eigenaarschap.

In een gelijkaardige studie³ (door Scott Ambler in 2009 gehouden via het internet en met 129 respondenten), kwamen continue integratie en de dagelijkse statusmeeting als de twee meest effectieve en eenvoudig aan te leren technieken naar voor. Gelijkaardige resultaten dus als in onze studie.

Op het gebied van actieve betrokkenheid van stakeholders en potentieel releasebare software lijken de ondervraagden in de Ambler studie minder overtuigd.



Figuur 8 Experimenteren met nieuwe software ontwikkelingstechnieken: effectiviteit en aanleergemak

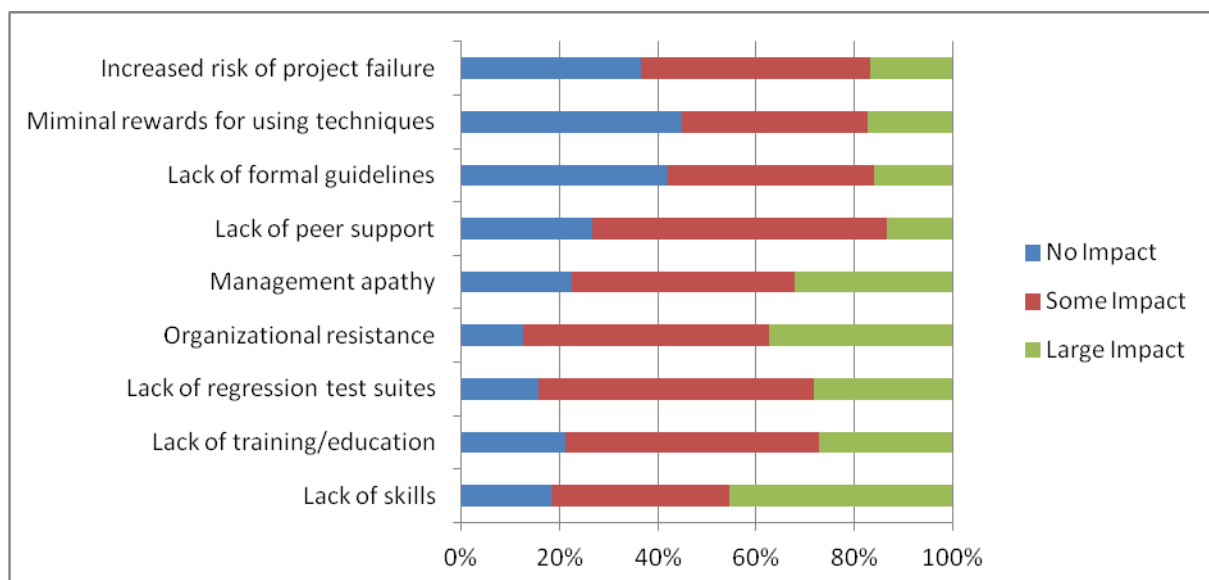
³ Scott Ambler. Amblysoft Agile Practices 2009 Survey, <http://www.amblysoft.com/surveys/practices2009.html>

4.2 BELEMMERINGEN VOOR INNOVATIE

We onderzochten ook wat een organisatie ervan weerhoudt om meer te experimenteren met technieken. De grootste belemmeringen om innovatie in het software-ontwikkelingsproces te brengen zijn een **gebrek aan kennis, training en opleiding bij de betrokkenen, alsook organisationele weerstand of desinteresse vanuit het management.**

De belangrijkste evaluatiecriteria voor verbeteringstrajecten in software-ontwikkeling zijn eerst en vooral klantentevredenheid, en daarna kwaliteit en onderhoudbaarheid. De aspecten kost en tijd, die zoals eerder in dit artikel beschreven het minst prioriteit krijgen bij standaardtrajecten, zijn ook niet de doelstellingen van verbeteringstrajecten.

Organisaties die vaker gunstige projectresultaten rapporteren (organisaties in categorie 'vaak' uit Figuur voor minstens 1 van de criteria) **verwachten de meerwaarde echter vooral op het vlak van tijd.** Men kan dus stellen dat deze organisaties er meer op gericht zijn pijnpunten aan te pakken.



Figuur 9 Belemmeringen voor innovatie

4.3 CONCLUSIE

Wanneer we peilen naar de mate waarin projecten de beoogde kwaliteitscriteria behalen, stellen we vast dat succesvol projecten opleveren binnen deze criteria zeker niet de standaard is. **Heel wat organisaties hebben moeite om tijd en budget onder controle te houden.** Het criterium klantentevredenheid scoort aanzienlijk beter.

Wanneer we gebruikte ontwikkeltechnieken en deze outputcriteria uitzetten tegen elkaar, dan stellen we vast dat meer **succesvolle organisaties zich voornamelijk onderscheiden door het toepassen van best practices in projectmanagement (inschatting, risicobeheer, reviews) en aandacht voor architectuurwerk. Agile development practices of projectmethodiek in het algemeen draagt niet bij aan het welslagen van projecten.**

Als redenen voor het **niet behalen van doelstellingen** worden vooral het wisselen tussen taken, en het gedeeltelijk opleveren van werk genoemd. Als **reactie** daarop wordt vooral de oplevering van functionaliteiten gespreid in de tijd, of worden niet-essentiële functionaliteiten geschrapt.

In welke mate onderscheiden organisaties die het best scoren zich van de anderen op het vlak van innovatie? Uit ons onderzoek blijkt dat de opgeleverde systemen vaker van hoge kwaliteit waren en een grotere aanpasbaarheid hadden.

Hieruit kunnen we afleiden dat een **hogere maturiteit met betrekking tot software-ontwikkeling ook vaker leidt tot het behalen van doelstellingen geformuleerd in innovatietrajecten.**