

Crossmedia technologie

Agile Software Development

Tom Peeters

Rapid software development

“Rapid development and delivery” is tegenwoordig een belangrijk requirement voor software systemen

- Businesses vragen een snelle ontwikkeling
- Men moet met verandering overweg kunnen
- Specification, design and implementation zijn “inter-leaved”
- Systemen worden ontwikkeld in verschillende versies waarbij stakeholders nauw betrokken worden



Agile methods

- Veel overhead zorgt voor ontevredenheid
 - 50% van de projecten vallen
 - 40% gaan over tijd en/of budget
 - Projecten worden nooit uitgevoerd als gepland
- DAAROM
 - Focus op code
 - Iteratieve benadering (change!)
 - Zo snel mogelijk een werkende versie. Van hieruit verder bouwen
- DOEL = Verminderen van overhead (minder gedetailleerde documentatie) + de mogelijkheid om snel te reageren op veranderende requirements



Wat maakt een project “Agile” (*principes*)

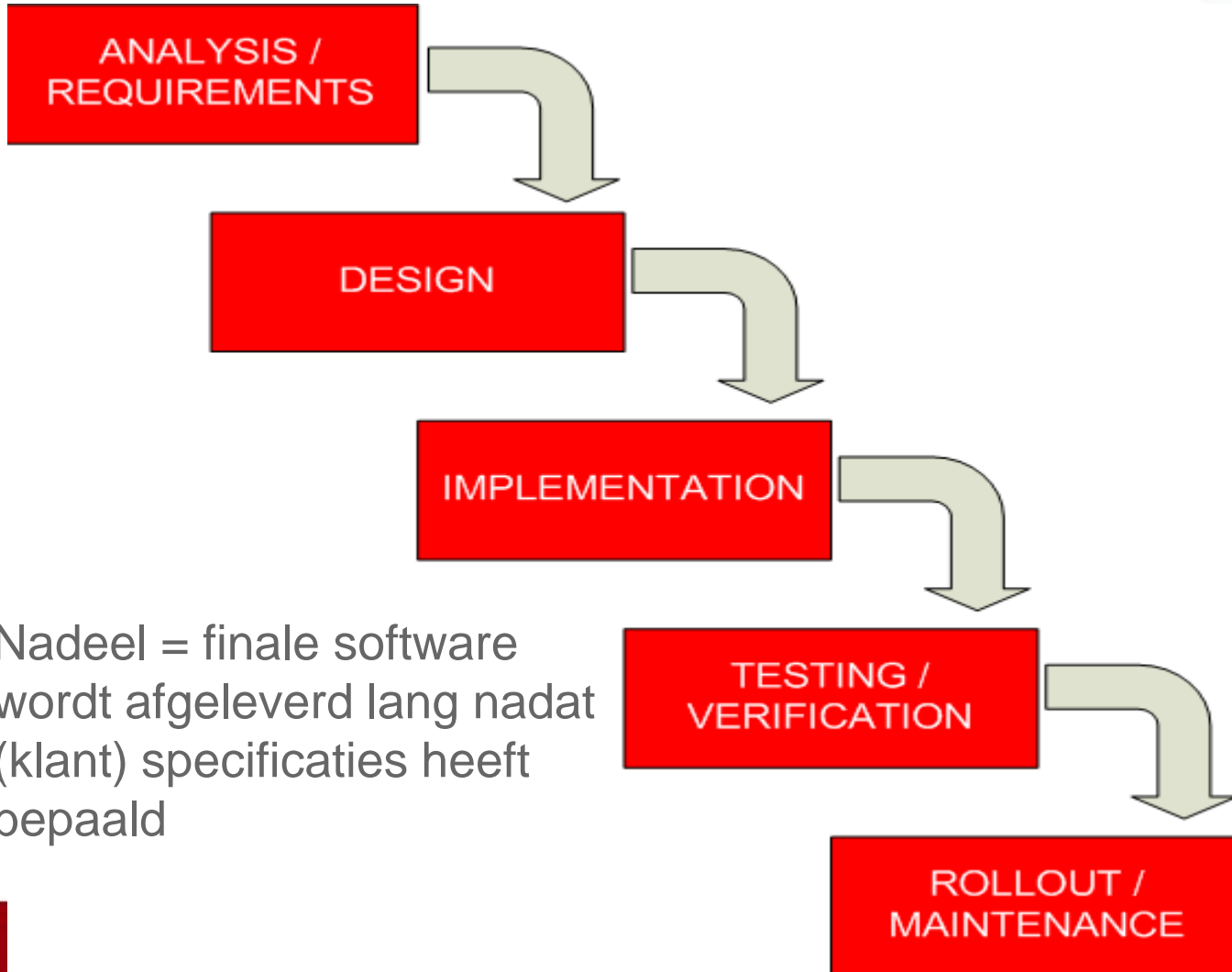
- Korte iteraties
 - Een deel van de software analyseren, ontwerpen, ontwikkelen, testen, accepteren en opleveren
- Samenwerkende teams
 - Mensen werken niet NA mekaar, maar MET elkaar
- Kleine eenheid van werk
 - Tijdens iedere iteratie wordt een gedeelte van de software opgeleverd (*user stories > work items*)
- Omgaan met veranderingen
 - Bij de start van een iteratie worden work items uit een lijst resterende work items gekozen die dan het belangrijkste zijn om te realiseren

Wat maakt een project “Agile” – *cont’d*

- Voortdurend plannen, meten, testen
- Vereenvoudigde communicatie
 - Post-its aan de muur (scrumboard), spreadsheets met overzicht van de *work items*



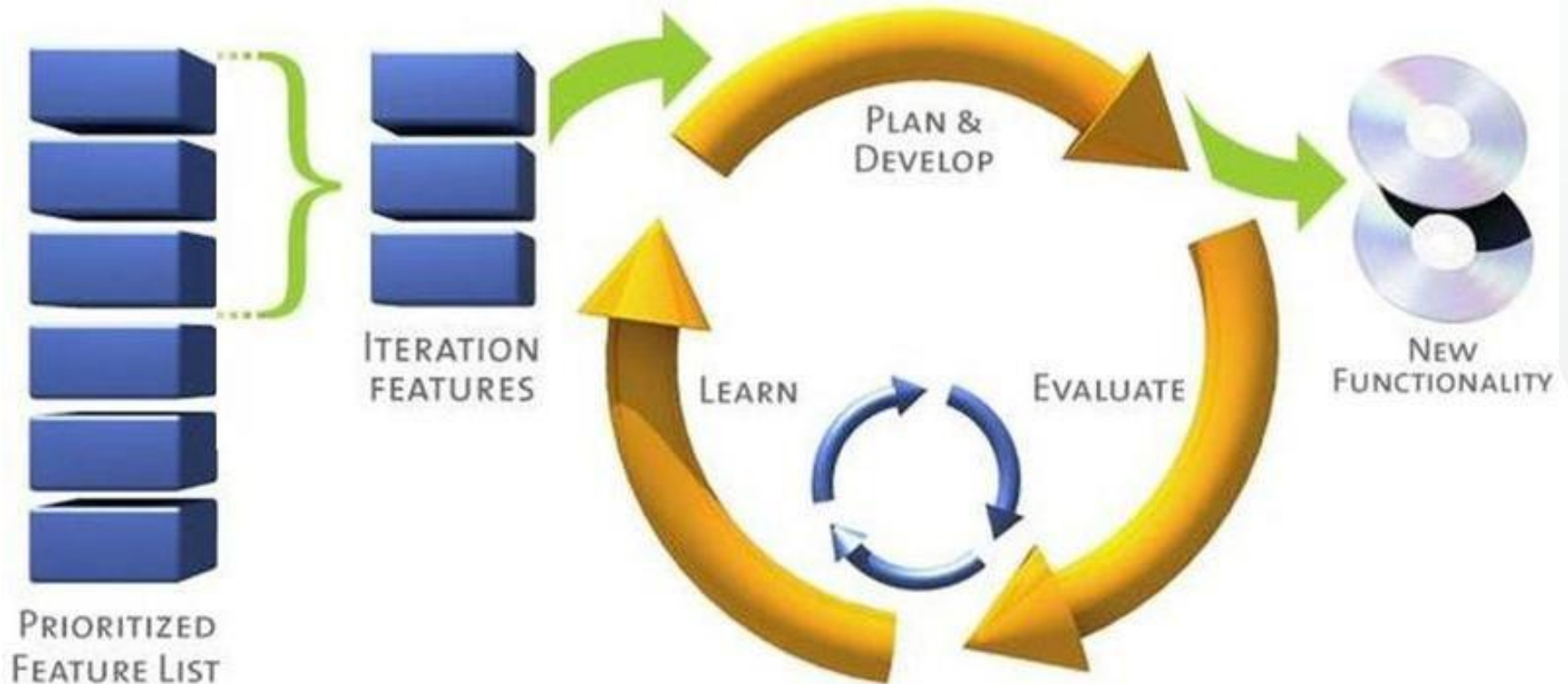
Traditioneel Project



- Nadeel = finale software wordt afgeleverd lang nadat (klant) specificaties heeft bepaald



Agile Project



Agile Project – terminologie

- Work items
 - Bepaald (klein) deel van het totaal project dat gerealiseerd kan / moet worden. (Epic = algemene workitem -> onderverdeeld in work items)
 - Product Backlog
 - Lijst met de te realiseren work items
 - Het team adviseert, maar de klant kiest!
 - Must Have, Should Have, Could Have, Want to have but won't have this time around
 - => Al het werk bij traditionele projecten die sequentieel verlopen (analyse – ontwerp – development – testing) worden per work item uitgevoerd
-
- Project gaat van start als product backlog bekend is (Hiervan hangt prijs af)

Agile Project : werkwijze

- Een Agile Project bestaat uit iteraties of sprints (scrum)



- Iteraties duurt 2 tot 4 weken
- Gekozen *work items* geprioriteerd en naar *sprint backlog* geplaatst die worden ontwikkeld

Problemen bij Agile

- Moeilijk om klant te motiveren om betrokken te blijven bij het project
- Team leden staan onder constante druk
- Prioriteiten toekennen is moeilijk door verschillende stakeholders (met elk een ander doel: marketing, gebruiker, manager..)
- Contracten: fixed pricing <> variable price



Waarom zorgt agile voor een snellere ontwikkel & deployment tijd?



Waarom zorgt agile voor een snellere ontwikkel & deployment tijd?

- Nadruk ligt bij de mensen en niet bij het proces
 - *Development team weet wat iedereen doet (dagelijkse stand-up meetings)*
 - *Geen overhead door formele communicatie*
 - *Team focust op ontwikkeling. Developer krijgen verantwoordelijke rol*
 - *Tijdswinst doordat er geen formele documenten worden verwacht*
- *Customer collaboration*
 - *In plaats van requirements discussie & afzonderlijke specificatie fase vragen developers onmiddellijk feedback aan de klant*
 - *Functionaliteit wordt sneller opgeleverd (iteratief proces)*
- *Responding to change over following a plan*
 - *Vraag naar verandering is onvermijdelijk*
 - *Planning vraagt tijd... Door te werken met sprints & prioriteiten kan men makkelijker overweg met verandering*

Agile Project : Iteratie / sprint

- Elke Iteratie bestaat uit 3 delen

• KICK OFF

- Hele team legt vast welke work items tijdens iteratie worden opgepakt
- Team commit met klant

• WERK

- Realiseren van afgesproken work items
- Realiseren = Analyse, ontwerp, implementatie, testen en acceptatie

• EVALUATIE

- Evaluatie met hele team + *stakeholders*
- Demo + werkwijze onder loep genomen



Agile Project : Iteratie > Kick off

- Start-up meeting van de iteratie
- Iedereen neemt deel
 - Opdrachtgever, klant
 - Project manager
 - Experts (testers, developers, analysten)
- Work items worden gekozen uit backlog, waarbij team advies geeft aan klant
- De lijst met gekozen work items voor een iteratie = iteration backlog of sprint backlog



Product backlog & iteration (sprint) backlog

#40 Relationale analyse van wireless parameters voor het maken van complexe grafieken voor Glen, test procedures voor ons model intelligenter te maken uit te werken

Todo Tasks | 0 Comments 👤 #E2

#38 Onderzoeken hoe realtime (<10s) metingen kunnen gebeuren

Uitzoeken of dit überhaupt kan. Fastpipes?

Todo Tasks | 0 Comments ☐ Inwave Deliverable 👤 #E1 / #E13 👤 GlenVR

#37 Invloed van driver/query snelheid meten: basic test

Testen wat impact is op generiek netwerk bij gebruik van 'tool'. Test gaat uit van: maximum 3 hops, 3x3. Tijd opmeten van binnenkomen PV & trenden van de latency.

Todo Tasks | 0 Comments ☐ Inwave Deliverable 👤 #E17 👤 GlenVR

#36 Bestaande compliancy test integreren (zonder dartaopslag)

Todo Tasks | 0 Comments 👤 #E12 👤 GlenVR

#35 Compliancy test

Procedure om te kijken of er überhaupt aan de gewenste commando's kan geraakt worden bij leverancier x

Todo Tasks | 0 Comments ☐ Inwave Deliverable 👤 #E17 👤 GlenVR

#41 refactoren van yaml emitter TCP

Done Tasks | 0 Comments 👤 #E8 / #E11

#45 contextuele informatie / interface

Todo Tasks | 0 Comments 👤 #E4

#43 ZMQ module voor node-webkit als stub beschikbaar maken in TypeScript omgeving

Reviewing Tasks | 0 Comments 👤 #E4

#42 documenteren van console parameters en functies hart/tcp/sqlite/zmq

Done Tasks | 0 Comments 👤 #E8 / #E11

#44 Grafieken parameter verloop over tijd

Doing Tasks | 0 Comments 👤 #E4

#39 Fastpipes onderzoeken: impact, nut, etc.

Doing Tasks | 0 Comments ☐ Inwave Deliverable 👤 #E1 / #E13 👤 GlenVR

#32 Procedure beschrijving?

Opzoeken hoe best procedures/Workflows bewaard worden (ism UA)



Evaluatie

- Iedereen is aanwezig
- Begin evaluatie meeting met:
 - Zijn de opgeleverde work items uitgevoerd
 - => Work items worden besproken en besloten of ze **definitief gereed** zijn
 - Is de werkwijze optimaal: wat gaat er goed, wat kan er beter
 - => Bij verbeterpunten worden hier direct acties gekoppeld



Rollen: De klant (stakeholder)

- Voortdurend in controle
- Start Project :
 - Klant legt scope en requirements op
 - Invulling Project backlog
 - Prioritiseren van work items
- Start iteratie:
 - Klant kiest work items
 - Uitwerken van *User stories*
- Einde iteratie:
 - De gerealiseerde work items worden door de klant definitief goedgekeurd



Rollen: Het Team

- Development Team
 - Diegene die project uitvoert
 - Team is zelforganiserend, tijdens kickoff wordt het werk verdeeld
 - Legt afspraken tussen klant vast: doelstelling, rollen, deadlines, werkwijze, acceptatiecriteria, beheerst verwachtingen (tijdens kickoff)



Rollen: Het Team – cont'd

- Ontwikkelaar
 - Betrokken bij requirements, inschatten van work items (discussie tijdens kick off)
 - Implementatie (code)
 - Architectuur
 - Pair Programming !
- Tester
 - Testen van work items
 - Uitschrijven & coderen van test plannen
 - Coderen
 - Uitwerken van deployment



(*Agile*) Requirements

- Traditioneel start het project met het vastleggen van alle requirements.
=> Nu start dit proces als de work items worden gerealiseerd
- Uitgewerkt '**Just In Time**'
- User Stories:
 - <http://www.mountangoatsoftware.com/topics/user-stories>
 - Minimale vastlegging van de requirements
 - Legt in 1 enkele zin vast wat de eindgebruiker met software wil bereiken
 - Vb. ALS lid van de bib
WIL IK een boek online kunnen verlengen
ZODAT ik niet naar de bib hoef
 - Voornaamste instrument in communicatie tussen klant en team
 - Deze User Stories verder uitwerken



De Sprint cyclus

- Sprints hebben een vaste tijdsduur: 2-4 weken
- Het startpunt is de product backlog (lijst van work items die moeten ontwikkeld worden)
- De SCRUM MASTER organiseert de dagelijkse meetings, houdt bij wat er gedaan moet worden, meet de progressive en geeft feedback aan de klant en het management
- Elke dag worden stand-up meetings gehouden waarbij iedereen van het team aan mee doet.
 - Info en progressive wordt gemeten tegenover de vorige dag
 - Problemen worden aangekaart
 - Planning vandaag wordt opgemaakt



User requirements worden als “stories” beschreven. Wat zijn de voor – en nadelen?



User requirements worden als “stories” beschreven. Wat zijn de voor – en nadelen?

Voordelen:

1. Stellen een echte situatie voor van gebruikers functionaliteit
2. Makkelijk te begrijpen
3. Stellen een “incrementele” functionaliteit voor: geeft aanleiding voor verdere discussie



User requirements worden als “stories” beschreven. Wat zijn de voor – en nadelen?

Nadelen:

1. Zijn niet compleet.
2. Focus op functionele requirements, en niet op non-functional reqs
3. Onderhoudbare code wordt moeilijk door het incrementele karakter



Betrokkenheid van de klant is essentieel,
MAAR:

ze worden deel van het development team
en raken vogelperspectief kwijt. Klant
wordt een collega van de ontwikkelaar

Hoe kan je dit vermijden?



Betrek meerdere stakeholders, zodat je meerdere ideeën/perspectieven krijgt.
(nadeel:meerprijs? Meerdere conflicten?)

Verander van klant (nadeel = duurt even vooraleer hij up-to-date is, met andere woorden productief. Je krijgt misschien conflicterende requirements)

Valideer suggesties met andere stakeholders



Oefening - Scrumdo

<https://www.youtube.com/watch?v=-XfCQhN0QRU>

Flip-chart classroom



ARTESIS PLANTIJN
HOGESCHOOL ANTWERPEN