

H04: valideren

Situering



Acceptance criteria

- GWT
- 3 amigo's
- Testplan

Valideren

- Begint met het bepalen van het succes
- 3 amigo's
 - Business: welke problemen willen we oplossen?
 - Development: hoe kunnen we het probleem oplossen?
 - Testing: lost deze oplossing het probleem kwalitatief op?
- Refinement sessies: user stories verduidelijken
 - Opsplitsen in kleinere stories
 - Acceptance criteria opstellen

Acceptance criteria

- Bepalen succes vd user story
 - Gebruikt door de testers om te valideren
- Gangbare manier om te definiëren: Given - When - Then
 - Behavior Driven (BDD)
 - User story als basis, meer details erbij voor GWT
 - Geeft aanzet naar functionele testen

Give/When/Then

Scenario: als een ingelogde user, wil ik kunnen inloggen op een website, zodat ik mijn persoonlijk profiel kan bekijken

GWT

- Given: ik ben een uitgelogde user, ik ben op de sign-in pagina
- When: als ik username & password invul met correcte credentials en ik klik op sign-in knop
- Then: systeem logt mij in

Testen

- Evolutie
 - Debuggen: onderdeel van programmeren
 - Zoeken naar fouten (los van programmeren)
 - Aantonen dat requirements voldaan zijn

Testen worden uitgebreider & gestructureerder

- Voldoen aan requirements
- Vinden van fouten
- Inzicht geven in kwaliteit
- Vertrouwen geven in kwaliteit van release
- Check op volledigheid & correctheid
- Advies geven over kwaliteit & risico's

Testen als vak

- Algemene kennis van informatietechnologie
- Kennis van infrastructuur, tools & ontwikkelomgeving
- Sociale vaardigheden
- Risk based testen
- Mensen, geld, tijd
- Verantwoorde wijze tests voorbereiden/uitvoeren
- Voldoende vertrouwen opbouwen om SW te accepteren

→ missing link tussen IT en business

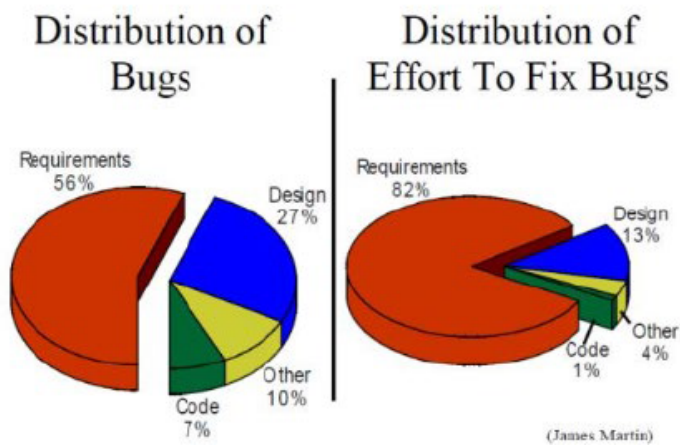
Waarom testen?

- Voor organisatie houdt elke wijziging een risico in
- Beperken van risico's: flexibiliteit verhogen
 - Softwareontwikkelp proces (Service Oriented Architecture)
 - Ontwikkelmethode die past bij de dynamiek van het bedrijf: agile development
 - Toepassen van controlemaatregelen: betrekken business bij opstellen/beoordelen systeemontwerp, uitvoeren van review/audits/...
 - Resultaatgedreven testen
 - Verificatie (het systeem goed bouwen)
 - Validatie (het goede systeem gebouwd is)
- Voorbeelden
 - Pentium-bug: FDIV bug (chip): niet genoeg precisie bij delen
 - Ariane 5: 2e vlucht: 2e trap niet losgekoppeld, oorzaak was SW-fout

Wanneer testen?

- In elke ontwikkelingsfase
- Inspanning: 30% tot 70% van het totaal
- Aandacht voor testen: afhankelijk van mogelijke gevolgen/risico's

Foutdistributie



Kost om defect te corrigeren

- Tijdens designfase: 1x
- Tijdens implementatie: 6.5x
- Tijdens testing: 15x
- Tijdens onderhoud: 100x

Schade aan klanten en gebruikers (dalende volgorde)

- Bedreigend voor veiligheid van menselijke wezens
- Aantasting ve essentiële organisatiefunctie
- Aantasting vd werking van firmware
- Aantasting ve essentiële organisatiefunctie, is een vervanging beschikbaar door alternatief systeem
- Aantasting vd goede werking ve software pakket voor business toepassingen
- Aantasting vd goede werking ve software pakket voor een private eindgebruiker
- Aantasting vd werking ve firmwaretoepassing maar zonder het volledige systeem te affecteren
- Brengt ongemakken voor de gebruiker mee maar verhindert hem niet om de systeemmogelijkheden te benutten

Schade aan de software-ontwikkelaar/organisatie

- Financieel verlies
 - Betaling schade voor fysieke blessures
 - Betaling schade aan organisaties voor de slechte werking van de softwareTerugbetaling van aankoopkosten aan klanten
 - Hoge onderhoudskosten voor het herstellen van falende systemen
- Niet-kwantitatieve schade
 - Negatieve impact op toekomstige verkoopcijfers
 - Substantiële terugval van de huidige verkoopcijfers

Toegevoegde waarde van testen

- Bijdrage aan het resultaat
- Oorzaak vd schade wegnemen
- Reductie van de impact
- Vertrouwen in het testobject
- Effectieve projectsturing

Binnen een SW-project: verschillende disciplines, belangen & aandachtspunten: projectmanager, programmeur, gebruiker, beheerder, business manager, tester

10 testprincipes

1. Focus op het resultaat
2. Bouw aan vertrouwen
3. Neem verantwoordelijkheid
4. Beheers het testvak
5. Sla bruggen
6. Test gefaseerd
7. Faciliteer de gehele IT-lifecycle
8. Geef overzicht en inzicht
9. Zorg voor herbruikbaarheid
10. Bedenk: testen is leuk!

1. Focus op het resultaat

- Focus op business resultaat
- Meer dan alleen bijdrage aan SW-kwaliteit
- Testen die bijdragen aan resultaten die de organisatie nastreeft
- Verificatie: aantonen dat systeem gebouwd is volgens systeemontwerp
- Validatie: inzicht geven in de mate waarin het systeem past binnen de bedrijfsprocessen

2. Bouw aan vertrouwen

- Vertrouwen opbouwen door intensief testen → minder kans op onopgemerkte fouten
- Probleem: beperkte resources (tijd, geld)
- Risicogebaseerd testen → belangrijke onderdelen grondiger testen
- Goede tester zoekt: fouten in andermans werk, goede elementen (positief kritisch)
- Goede tester betreft actief belanghebbenden bij het bepalen van risicogebieden, aandachtspunten → transparantie/volledigheid/expertise
- Verlies geen tijd en energie met het maskeren van fouten
- Beter op tijd een probleem
- Ervaring: bereidheid/begrip/aanvaarding van de consequenties indien dit tijdig gesignaleerd wordt

3. Neem verantwoordelijkheid

- Juiste testen voorbereiden
- Rekening houden met risico's
- Testdekking
- Overzicht testproject incl. controle
- Sterk betrokken bij IT-lifecycle
 - Slechte inventarisatie van de klantwens
 - Ondoordacht ontwerp
 - Onvoldoende kwaliteit programmeerwerk
 - Slecht configuratie- en versiebeheer

4. Beheers het testvak

- Testen is meer dan 'knoppen drukken'
- Goede kennis van systeemontwikkeling, testmethodieken
- Goede kennis van processen in organisatie
- Gesprekspartner voor de gebruikers/managers
- Bruggen slaan

5. Sla bruggen

- Veranderende requirements
- Inleving (belanghebbende)
 - De tester kan aan de ontwikkelaar uitleggen hoe de belanghebbenden naar het systeem kijken
 - De tester begrijpt de ontwikkelaar voldoende om samen met hem de bevindingen te analyseren

6. Test gefaseerd

- Gefaseerd testen resulteert in de beste SW-kwaliteit
- Duidelijke project aanpak, planning
- Logische volgorde
- Afdekking van de belangrijkste risico's
- Testproces
 - Strategie sluit aan op SW-ontwikkelp proces
 - KSF worden verdeeld over verschillende testsoorten (planning)
- Voordelen
 - Elke partij kan zijn voorbereidend werk doen
 - Entry- en exitcriteria(!)
 - Stap-voor-stap wordt duidelijk of het systeem aan de verwachtingen voldoet

7. Faciliteer de gehele IT-lifecycle

- Ontwerpfase
 - De producten toetsen op hun testbaarheid
 - Nagaan of het ontwerp aangeeft hoe het beoogde resultaat behaald gaat worden
- Ontwikkelfase
 - Hoe vroeger de fouten gedetecteerd worden hoe beter
 - Moduлетesten, module-integratietesten
- Beheerfase
 - Delen van de testware
 - Goede overgang van projectfase naar beheerfase

8. Geef overzicht en inzicht

- Systeem = veel functies
- Per functie worden groot aantal testen uitgevoerd
- Testresultaten
 - Functietesten
 - Afdekken van risico's, aandachtspunten, kwaliteitsaspecten

9. Zorg voor herbruikbaarheid

- Tijdens het gebruik van de SW zullen nog fouten optreden
- Herbruikbare testset → automatiseren?
- Testtraject kost tijd en geld → herbruikbaarheid = kostenbesparen
- Risicoanalyse actualiseren

10. Bedenk: testen is leuk!

- Toegevoegde waarde binnen het SW-ontwikkelp proces
- Verantwoordelijkheid voor het in productie gaan van een systeem
- Flexibel en creatief zijn
- Specialisatie is mogelijk: testautomatisering, performance-testen en security-testen

Soorten testen

Moduletest (MT) & module-integratietesten (MIT).

- Geautomatiseerd
- Whitebox-testen
- Unit-testen, component-testen, programma-testen

Systeemtest (ST).

- Een heel systeem testen
- Blackbox-test
- Werkt het systeem conform het functionele ontwerp
- Bouwende partij

Functionele acceptatietest (FAT).

- Een heel systeem testen
- Opdrachtgever
- Verificatie a.d.h.v. systeemeisen en functioneel ontwerp

Gebruikersacceptatietesten (GAT).

- Validerende test
- Representatieve scenario's uit het dagelijkse werk van de gebruikers
- Gebruiksvriendelijkheid, bruikbaarheid, ...

Productieacceptatietesten (PAT).

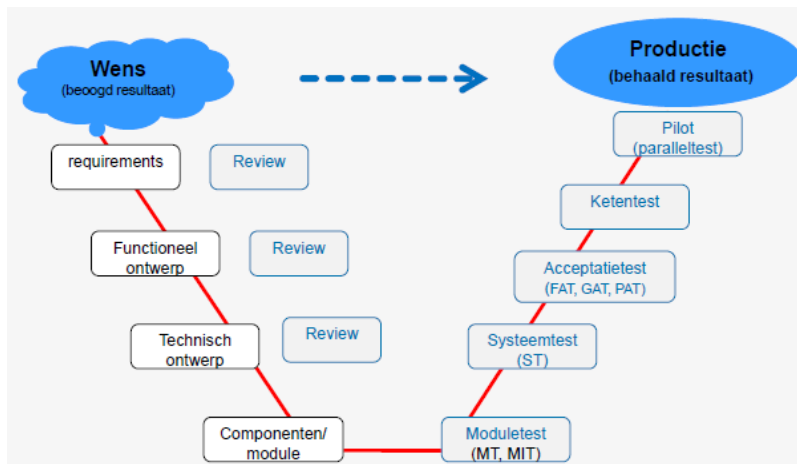
Ketentesten

- Samenwerking met de aanliggende systemen
- Integratietest

Pilot

- Parallel testen met bestaande systeem

Testen - vervolg



Verschillende testmethodologieën

- Agile testing
- Scripted testing
- Exploratory testing
- Rapid software testing

Afhankelijk van

- Project organisatie
- 'Application criticality'
- De testers

Testplan

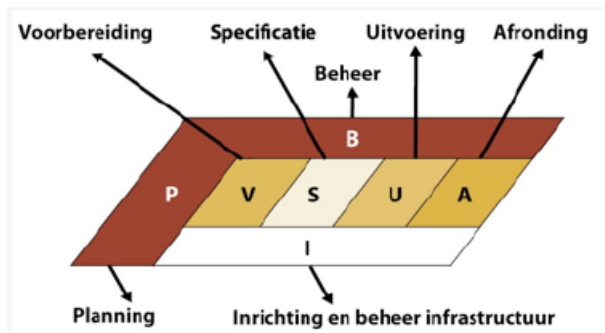
- Begin ve testtraject: testplan opstellen
- Doel vh testplan: aanpak vh testen binnen het programma/project/release/change te beschrijven
- Testplan opgesteld door de verantwoordelijke voor het testtraject → kan zijn de testmanager, de testcoördinator, de functioneel beheerder of ieder ander die de opdracht krijgt en aanvaard

Testplan

- Aanleiding → waarom
- Aanpak → hoe
- Benodigde mensen & middelen → wie en wat
- Benodigde planning → wat
- Op te leveren producten → wat

Onderwerpen die een rol spelen bij het opstellen van een testplan

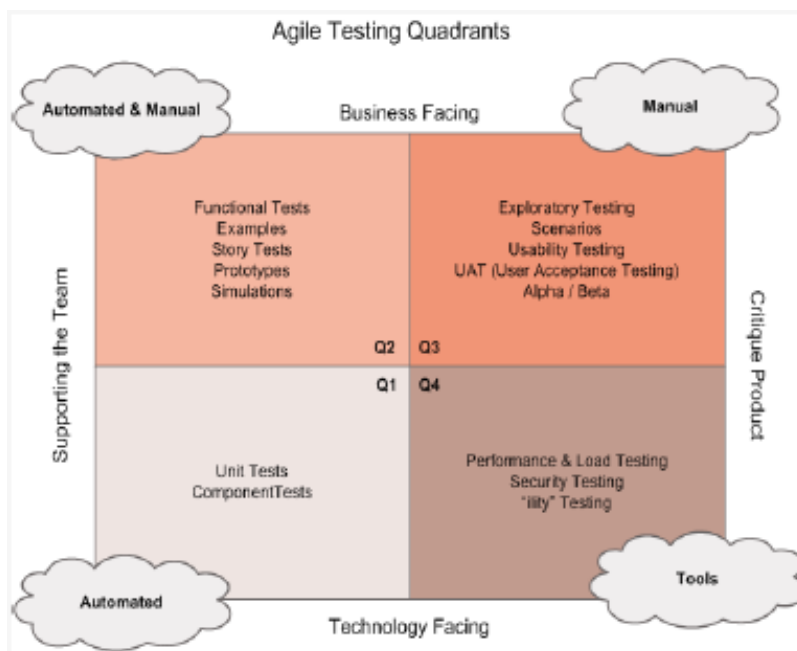
- Teststrategie
- Randvoorwaarden en uitgangspunten
- Entry en exit criteria
- Testsoorten entestvormen
- Testontwerpstechnieken



Wie voert de testen uit?

- Testers
- Programmeurs
- Business Analisten
- Ontwerpers (designers)
- Eindgebruikers
- Dedicated testing professionals
 - Opgeleid
 - Ervaren
 - 'Carriere'
 - Specialisatie
 - Security testers
 - Performance testing
 - Test automators
 - Embedded software testers
- Testing is een erkend beroep

Testautomatisering



Kwaliteitsborging

- Controle op basis van de opgeleverde tussenproducten of de vooropgestelde kwaliteit wordt bereikt
 - Vroegtijdig vaststellen van afwijkingen
 - Vroegtijdig herstellen
 - Kostenbesparend (Boehm)
- Hoe?
 - Collegiale reviews
 - Expert reviews
 - Inspecties
 - Walk-throughs
 - Monitoring (opdrachtgever) = audit

6 beoordelingsgrondslagen om een product te beoordelen



Toetstechnieken

- Collegiale review
- Expert view
- Structured walk through
- Inspectie

Collegiale review

- Collega's, die hetzelfde werk doen, beoordelen collega's
- Collega's zijn goed op de hoogte van omgeving, problematiek, technische achtergrond
- Minder vertrouwen in "vreemden" (auditor, inspecteur)
- Positief: collega's leren van elkaar

Risico's

- Te informele organisatie
 - Duidelijkheid geven over de review
 - Welk product, aspecten, hoeveel tijd, hoe bevindingen worden vastgelegd en hoe deze gecommuniceerd worden, aangeven wat prioritair is (review of eigen dagelijkse taak)
 - Bijzondere maatregelen
 - Training, uitvoeren van een pilot, ondersteuning bij uitvoering
 - Doel: collega's de meerwaarde van collegiale reviews laten onderkennen

- Geen eenduidige beoordelingsgrondslag
 - Vooraf bepalen welke aspecten in de review moeten belicht worden
 - Enkel inhoud?, voorgeschreven standaarden gehanteerd?, taalfouten corrigeren?, controle of vertaalslag correct gebeurd is? (Bv. FO naar TO), checklist?
 - Heldere beoordelingslag = verbetering van de kwaliteit
- De opsteller van het product en de reviewer dienen over voldoende communicatievaardigheden te beschikken
 - Collega's zien elkaar als concurrenten
 - "Afkragen" van het product wordt beschouwd als oncollegiaal gedrag
 - "Oncollegiaal gedrag moet voorkomen"
 - Maker van het product mag collegiale review niet zien als aanval op zijn werk
 - Doel van de review: het product verbeteren, niet het onderuit halen
 - Gedeelde verantwoordelijkheid
 - Product is niet van de maker alleen maar is een projectaangelegenheid
 - Kwaliteitsverbetering is een organisatiebelang
- Te vergevingsgezind
 - De reviewer meldt de geconstateerde afwijkingen niet

Conclusie

- Collega's beoordelen elkaars werk
- Zeer informeel
- Vooral gericht op inhoud
- Minder gericht naar vorm, toepassen van procedures
- Collega's leren van elkaar
- Voorwaarde: vertrouwen tussen de collega's

Expert review

- Expert die geen deel uitmaakt van het team
- Resultaten blijven intern
- Iets formeler van aard
- Gebruik van checklist, bevindingenlijst, rapport verhogen de kwaliteit van de review
- Minpunt: de expert kent de specifieke aspecten van de omgeving waarin de maker van het product werkt niet
- Pluspunt: de expert werkt onbevooroordeeld

Structured walkthrough

- De auteur leidt de groep door een document/product en licht achterliggende gedachte en keuzes toe
- Doel: consensus bereiken over de inhoud
- Auteur = actieve rol

Inspectie

6 fasen (Fagan)

- Planning
- Kick-off
- Voorbereiding
- Groepsmeetings
- Aanpassing product
- Afsluiting

Planning

- Toetsing entry criteria (=eisen waar een product aan dient te voldoen om geïnspecteerd te worden) → checklist
- Strategiebepaling: wie inspecteert welke aspecten?
- Verzamelen documenten: referentiekader opbouwen
- Benaderen inspecteurs

Kick-off

- Meeting: maker, inspecteurs, aspecten, belang

Voorbereiding

- Alle betrokkenen zoeken naar fouten

Groepsmeetings

- 3 meetings: Bevindingen-registratiemeeting, discussiemeeting, oorzaakanalyse
- Kwaliteitsmanager leidt vergadering

Aanpassing product

- Op basis van de bevindingen gebeuren aanpassingen
- Onafhankelijke functionaris (kwaliteitsmanager) checkt of alle bevindingen goed verwerkt zijn
- De maker blijft eindverantwoordelijke (hij kan sommige aanbevelingen bewust naast zich neerleggen)

Afsluiting

- Korte rapportage: auteur, inspecteurs, management