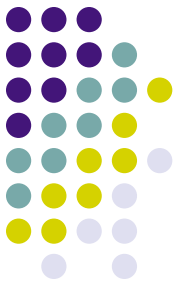


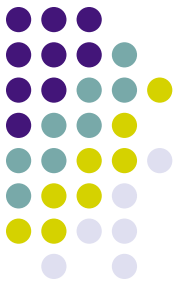
Software Engineering



1

Introductie in Software Engineering

Academische Integriteit



Software Engineering is een activiteit waar samenwerking voorop staat. Je wordt aangemoedigd om samen te werken, maar ...

- Sommige taken moeten alleen worden uitgevoerd.
- Geef altijd credits aan je bronnen en groepsgenoten.

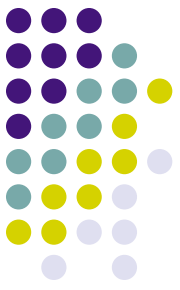
Goede professionele praktijk: maak gebruik van de expertise van anderen en bouw voort op eerder werk, **lettend op copyright.**

Onethisch gebruik (plagiat): gebruik van andermans werk **zonder te letten op copyright.**

Beoordeling



Project (groepswerk)	60%
Project (individueel)	40%



Feedback over de projecten

Er zijn 3 momenten waarop er geplande feedback zal zijn.

Projectplan

Als het projectplan af is zal dit beoordeeld worden en er wel of geen fiat gegeven worden om verder te gaan.

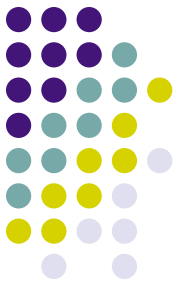
Prototype

Na de eerste implementatie van het prototype vindt presentatie plaats en klassikale evaluatie.

Eindproduct

Het eindproduct wordt gepresenteerd en beoordeeld.

Doel van het project



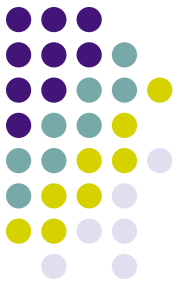
We gaan ervan uit dat je voldoende kennis hebt. Je weet voldoende van computers, kunt redelijk programmeren en en wilt meer leren van het werk.

Na afloop van dit project heb je zicht op hoe software productie verloopt waar succes maar ook verlies miljoenen Euro' s kost.

Misschien vind je er later werk in en betekent het je inkomen!

Fouten mogen nu gemaakt worden want van fouten kun je leren.

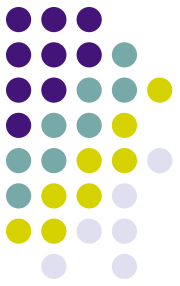
Projecten



De cursus is gebaseerd op de projecten

- Echt project voor een echte klant die van plan is de software te gaan gebruiken.
- Selecteer je eigen project, met een methode van ontwikkeling.
- Projectteam bestaat uit 3 personen.
- Maken projectplan en nagaan van uitvoerbaarheid
- 3 Tussentijdse presentaties en verslagen:
 - of: requirements, ontwerp, eindproduct
 - of: eerste iteratie, tweede iteratie, eindproduct

Selectie van een project



Kijk op de website

- Enkele projectideeën staan vermeld op de website
- Je wordt aangemoedigd om een eigen project te vinden.

Neem contact op met mogelijke klanten

- Probeer te achterhalen wat zij willen
- Schat de mogelijkheden van het project in.
- Discussieer over de te nemen besluiten met de klant.

Maak een projectteam

- Geef de samenstelling door op p.geelen@groenewoud.nl
- Zorg voor evenwichtige verdeling.

Overwegingen bij project



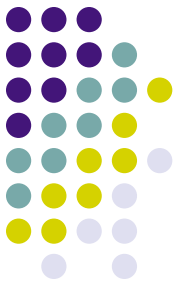
Projecten

- Onderwerp moet een productiesysteem opleveren.
- Er moet regelmatig contact zijn met de klant.

Team

- Teams moeten samengesteld zijn uit personen met verschillende sterke kanten -- organisatie, technisch, documentatie, programmeren, etc.
- Stel een voorzitter aan die het geheel coördineert.

Varieteit aan Software



Voorbeelden

<i>Real time:</i>	luchtvaart controlesysteem
<i>Embedded systemen:</i>	digitale camera, GPS, iPod
<i>Data processing:</i>	telefoonrekening, pensioenen
<i>Informatie systemen:</i>	websites, digitale bibliotheken
<i>Sensors:</i>	weerstation
<i>Systeem software:</i>	besturingssystemen, compilers
<i>Communicatie:</i>	routers, telefoonswitches
<i>Office pakketen:</i>	tekstverwerking, videoconferentie
<i>Wetenschappelijk:</i>	simulatie, weersvoorspelling
<i>Grafisch:</i>	film maken, ontwerp
<i>etc., etc., etc.,</i>	



Software is duur

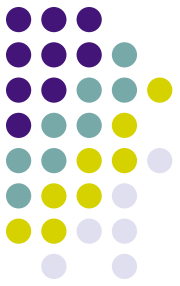
Software is duur!

De bulk aan kosten zijn salarissen (jouw salaris)!

Elk software project heeft een evenwicht tussen:

- Functionaliteit
- Resources (kosten)
- Tijdlijnen

Software is duur



Wie betaalt het geld?

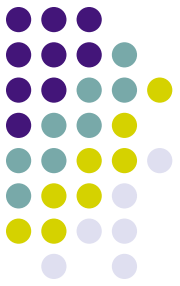
Wat wil die persoon of organisatie?

- Wat is succes?
- Wat is mislukking?

Technici hanteren vaak andere criteria voor succes als de mensen die het voor het zeggen hebben in de organisatie.

Voorbeeld: Vroege Unix workstations, Sun en IBM

Klant



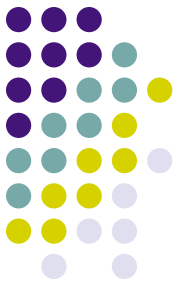
Klant

De klant stelt resources ter beschikking en verlangt daarvoor een software product terug.

De klant is vaak een lid van de organisatie die het geld ter beschikking stelt. Het succes van de klant in de organisatie hangt af van het succes van het software project.

Klanttevredenheid bepaalt in hoge mate het succes van het software project.

Vraag: Wie is de klant van Microsoft Excel?



Klant: Categorieën producten

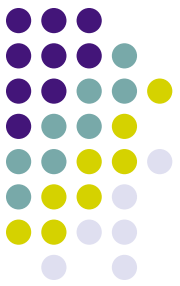
Klanten en categorieën software producten:

- Algemeen (b.v., OpenOffice Calc)
- Packages (b.v., Mathematica)
- Aangepaste versies van algemene packages
- Demonstratie, prototype, onderzoek, ...

Wie is de klant voor welk product?

Wie geeft het geld?

Wat willen ze?



Software is riskant

Hoe kun je risico's managen?

- *Veel software wordt nooit gebruikt (misschien 50%)!*
- *De meeste software ontwikkelprojecten hebben grote problemen!*

Wat is de schuld als de ontwikkeling van de software:

te laat is?

het budget heeft overschreden?

een niet werkend of een buggy programma oplevert?

Voorbeelden:

auto anti-blokkeersysteem (geen bugs toegestaan)

webbrowser in mobiele telefoon (geen vertragingen)

Software is riskant



De meeste software projecten falen omdat de software ontwerpers de verkeerde software bouwen!

- Probeer goed te begrijpen wat de klant van de software verwacht
- Probeer te begrijpen wat de organisatie van de klant verwacht
- Geef technisch inzicht en suggesties, maar onthoud:

Klanttevredenheid bepaalt in hoge mate het succes van het software project.

Teams



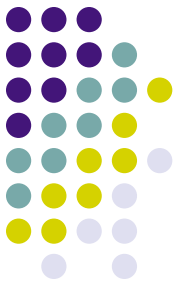
De meeste softwareontwikkeling vindt plaats in **teams**

- De efficiëntie van een team bepaalt het succes

De meeste grote software projecten worden **uit oudere producten gemaakt**

- Het komt zelden voor dat een groot programma from scratch wordt opgebouwd
- Voortbouwen op het werk van anderen is een fundamentele toepassing van software ontwikkeling

Observaties aan grote projecten



- Een NSG SE project duurt 60 SLU per persoon per jaar. Een groot project kan 100 tot 1000+ manjaren in beslag nemen.
- Elk groot project wordt door vele mensen uitgevoerd met in samenstelling wisselende groepen.
- Voordat een groot project is afgerond zijn de requirements vele malen gewijzigd.
- Geen enkel groot systeem is ooit afgerond.

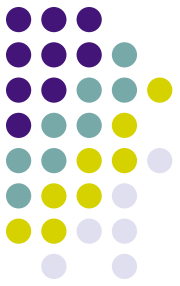
Voorgaande ervaringen (jouw)



Je achtergrond

- Grootste programma dat je geschreven hebt?
- Grootste project waaraan je hebt gewerkt?
- Grootste projectteam waarin je hebt gezeten?
- Langste project waarin je hebt gezeten?
- Welke mensen hebben jouw werk gebruikt?

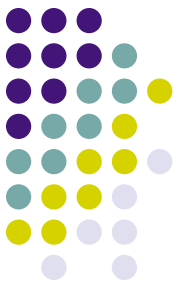
Toekomstige ervaringen



Wat doe je over een jaar?

En wat over 10 jaar?

Typische carrierepad in computerwetenschap combineert technisch werk met verschillende graden van project management, marketing, ondernemerschap, etc.



Cursus thema' s: risico' s

Risico (door de ogen van een manager)

- **Problemen**

 - Overschrijding budget

 - Te late oplevering

 - Doet niet wat het moet doen

- **Nooit gebruikt**

 - Doet het verkeerde

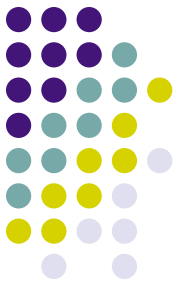
 - Moet worden veranderd

 - Gebruikers willen er niet aan

 - etc.*

***Mislukkingen van software ontwikkelprojecten
kunnen bedrijven failliet laten gaan!***

Cursus thema's: transparantie



Transparantie (door de ogen van een manager)

- **Probleem**

Afhankelijk van anderen voor de rapportage over de voortgang en de eventuele problemen

- **Software ontwikkelaars**

Kunnen voortgang moeilijk inschatten

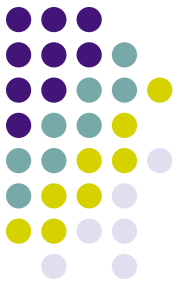
Vaak te optimistisch

Beschouwen rapporteren als tijdverlies

etc.

De verantwoordelijken moeten op de hoogte zijn van wat er gebeurt!

Cursus thema's: het proces

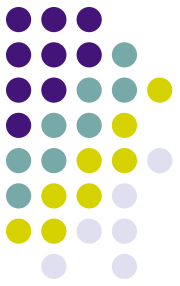


Het proces in grote software projecten

- Software als product
 - Kwaliteit, prestatie, bruikbaarheid
 - onderhoudbaarheid, ontwikkeling
- Scheiding van requirements en ontwerp
- Projectmanagement
 - Personeel management
 - Economische, wettelijke en sociale factoren
- Ontwikkelproces
 - Sequentieel (waterval model)
 - Iteratieve verfijning

Iedereen heeft slechts een deel van de kennis van een groot project !

Cursus thema's: schaal



Technieken voor grote en erg grote systemen

- Software ontwerp
 - Software architectuur
 - Objectgeoriënteerd ontwerp
- Afhankelijke systemen
 - Betrouwbaarheid
 - Verificatie
- Oude systemen

Wat is goede software?



Algemene eigenschappen

Bruikbaarheid

Onderhoudsvriendelijk

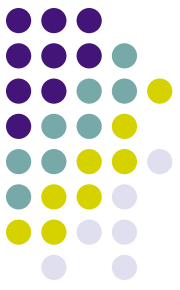
Betrouwbaarheid

Efficiënt

Goede software producten vergt goed programmeren,

maar ...

Programmeervaardigheid is het middel om het eind te bereiken, **niet het einde** zelf.



Vakmanschap van software ontwikkeling

Software producten zijn zeer gevarieerd

- > Klant requirements zijn erg verschillend
- > Er bestaat geen standaard proces voor software engineering
- > Er bestaat geen beste programmeertaal, besturingssysteem, platform, database systeem, ontwikkelomgeving, etc.

Een bekwame software ontwikkelaar kent verschillende benaderingswijzen, methodes en tools. Het **vakmanschap** van software ontwikkeling is de juiste methodes voor elk project te kiezen en ze effectief toe te passen.

Professionele verantwoordelijkheid



Organizaties stellen vertrouwen in software ontwikkelaars:

- Competentie: Software that does not work effectively can destroy an organization.
- Vertrouwen: Software developers and systems administrators may have access to highly confidential information (e.g., trade secrets, personal data).
- Legale omgeving: Software bestaat in een complex wettelijke omgeving (b.v., intellectueel eigendom, stuitende content).
- aanvaardbaar ge-(mis)bruik: misbruik van de computer kan een organisatie platleggen (b.v., een Internet worm).

Volgende stappen



- Lees info op de website.
- Begin met de planning van de projecten en de vorming van projectteams.