



Requirements ervaringen in de embedded wereld

Process Vision

Willem van den Biggelaar



Process Vision

- ◆ Projectbegeleiding van opstart tot en met eind-evaluatie als Quality Officer
- ◆ Coördineren van proces verbeterings trajecten
- ◆ Project assessments
- ◆ Aannames assessment door middel van rollenspelen
- ◆ Trainingen/workshops op de procesgebieden:
 - Requirements management
 - Project planning/tracking
 - Reviewing and inspecties
 - Validatie en testen



Willem van den Biggelaar

- ◆ HTS electrotechniek achtergrond
- ◆ 14 jaar ervaring in de embedded software waarvan
 - eerste 10 jaar in de software ontwikkeling
 - laatste 4 jaar als SPI consultant / Quality officer / trainer
 - docent test methodieken ontwerpersopleiding TUE



Inhoud

- ◆ Requirements Management of Engineering?
- ◆ Trends embedded software wereld
- ◆ Praktijkervaringen



Requirements Engineering (RE)



- ◆ Vragen komen op als
 - Hoe komen we aan onze requirements?
 - Hoe schrijven we ze op?
 - Zijn onze requirements wel goed?
consistent, meetbaar, traceerbaar, maakbaar, compleet,.....

*Het vakmanschap van het maken
(achterhalen/definieren) van requirements*



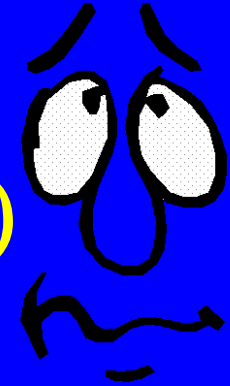
Slechte Requirements Engineering



- ◆ Enkele mogelijke gevolgen
 - veel wijzigingen en discussies tijdens ontwikkeling
 - onvolledige/overvolledige producten
 - onzuivere en instabiele planning



Requirements Management (RM)



- ◆ Vragen komen op als
 - Zijn alle partijen het eens en begrijpt men elkaar?
 - Hoe bevriezen we onze requirements?
 - Hoe wijzigen we bevroren requirements?
 - Hoe houden we onze tussenproducten en plannen consistent met deze wijzigingen?

*De organisatie, communicatie
en het beheer van requirements*



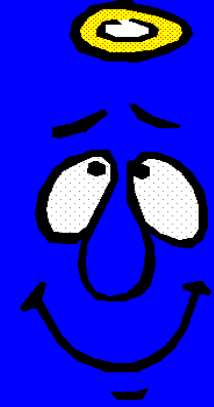
Slecht Requirements Mangement



- ◆ Enkele mogelijke gevolgen
 - onduidelijke afspraken tussen klant en project
 - requirements blijven zweven
 - wijzigingen worden niet of slecht geïmplementeerd
 - inhoud van eind product is niet meer duidelijk, wat hebben we nu opgeleverd?



Uitspraak

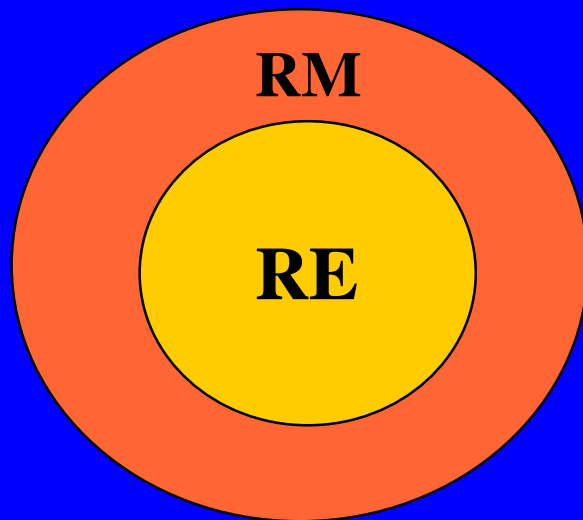
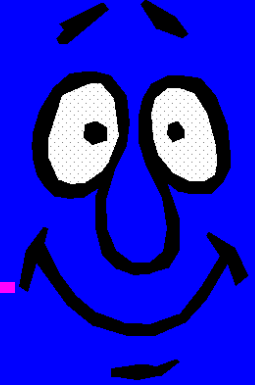


“Wij zijn een CMM level 2 organisatie, dus we hebben onze requirements goed op orde”

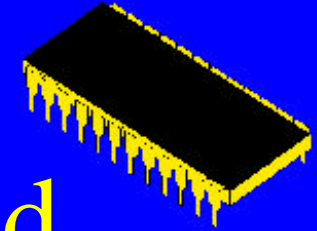
Oftewel het
betonnen zwemvest
syndroom



Tussenconclusie



Start geen
Requirements Management
verbetertraject
zonder ook
Requirements Engineering
te verbeteren.



Trends embedded software wereld

- ◆ Time-to-market (-production) steeds korter
- ◆ Functionaliteit complexer/uitgebreider
- ◆ Voortdurend nieuwe technologieën
- ◆ Producten opener dan voorheen
- ◆ Multi-disciplinaire ontwikkeling
- ◆ Diverse soorten klanten
(service, onderhoud, productie, eind-klant)



Praktijkervaringen Organisatie



- ◆ Bepaal wie je klanten/opdrachtgever(s) zijn
- ◆ Schakel al betrokkenen (klant/opdrachtgever/ontwikkelteam) zo snel mogelijk in (bijv. via brainstorm sessies incl. facilitator)
- ◆ Spreek schrijfmethoediek af met ontwikkelteam èn klant: beiden moeten het begrijpen.

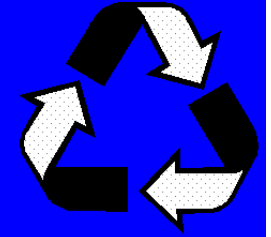


Praktijkervaringen **Kwaliteitseisen**

- ◆ Besteed extra aandacht aan niet functionele requirements: maak deze ‘gevoelseisen’ hard en meetbaar
- ◆ Indien klant het niet (goed) weet, stel zelf minimale (ervarings)eisen op en stem af



Praktijkervaringen **Hergebruik**



- ◆ Onderzoek hergebruik
 - lange termijn requirements + delta's per project
 - extra aandacht aan architectuur
- ◆ Stel je hebt al bestaande 'proza'
 - begin simpel bv start met labelen/nummers
 - discussies volgen vanzelf



Praktijkervaringen **Productgrootte**



Driver	Doorlooptijd	Inspanning (kosten)	Aantal productfouten
Productgrootte	+	+	+
Productiviteit	-	-	-
Tijdsdruk	-	+	+

*30-50% van automatiseringsproducten
bezit ongebruikte functionaliteit*



Praktijkervaringen Productgrootte



- ◆ Minimaliseer productgrootte
- ◆ Klassificeer requirements samen met klant
 - “must have”
 - “should have”
 - “nice to have”
 - “won’t have”



Praktijkervaringen **Proces**

- ◆ Haal vage/onduidelijke/open eisen (tbd, tbs) naar boven via open issue lijsten en spreek proces erover af
- ◆ Laat wijzigingsbeheer lopen via stuurgroep
 - architect: geeft technische impact
 - projectleider: geeft budget/tijd impact
 - klant: beslist uiteindelijk



Waarom projecten mislukken

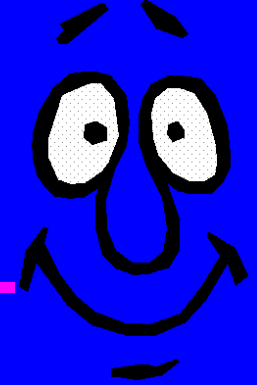


1. Incomplete requirements	13.1%
2. Lack of user involvement	12.4%
3. Lack of resources	10.6%
4. Unrealistic expectations	9.9%
5. Lack of executive support	9.3%
6. Changing requirements/specs	8.7%
7. Lack of planning	8.1%
8. Didn't need it any longer	7.5%

Bron: Standish group



Project succes factoren



1. User involvement	15.9 %
2. Mangement support	13.9 %
3. Clear statement of requirements	13.0 %
4. Proper planning	9.6 %
5. Realistic expectations	8.2 %
6. Smaller project milestones	7.7 %
7. Competent staff	7.2 %
8. Ownership	5.3 %
9. Clear vision & objectives	2.9 %
10. Hard working, focussed staff	2.4 %
11. Other	13.9 %

Bron: Standish group



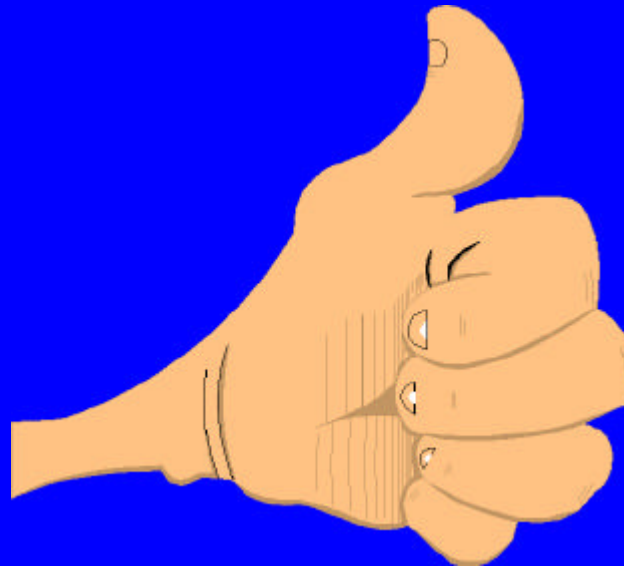
Conclusies

- ◆ Verbeter naast je proces ook je vakmanschap
- ◆ Weet wie je klanten zijn; betrek ze er a.s.a.p. bij en houdt ze erbij via de stuurgroep
- ◆ Denk aan hergebruik en minimalisatie
- ◆ Vergeet de kwaliteits eisen niet
- ◆ Breng open issues in kaart; beheer ze net als de wijzigingen op de bevroren eisen



Tot slot

Zijn voorgaande conclusies ook buiten de embedded wereld te gebruiken



Zeker weten