

# Vaatimusten ja testauksen yhteys

International Merito Forum Oy -tilaisuus / 09.06.2005

SoftQA  
Pekka Mäkinen  
<http://www.softqa.fi/>  
Pekka.Makinen@softqa.fi

# Kaksijakoisuus: vaatimukset vs. testaus

- ”Laatu on vaatimustenmukaisuutta” (Philip Crosby)
  - Minkälaiset ovat hyvät vaatimukset?
  - Miten vaatimustenmukaisuus tarkistetaan?
- Asioita voidaan tarkastella näiden kysymysten mukaan kahdesta suunnasta:
  - Vaatimusten testaus ja laadunvarmistus
  - Vaatimusten käyttö testauksessa ja laadunvarmistuksessa.
- Loppujen lopuksi nämä kaksi nivoutuvat yhteen, koska testauksen suunnittelu testaa myös vaatimukset ja hyvät vaatimukset ovat pohjana testaukselle.

# Mitä on vaatimustenhallinta?

- ~~Vaatimustenhallinta on työkalun käyttöä~~
- ~~Vaatimustenhallinta on oikeiden vaatimusten tuottamista~~
- ~~Vaatimustenhallinta on projektitoiminnan osa, joka tähtää laadun varmistamiseen~~
- ~~Vaatimustenhallinta on tylsää~~
- ~~Vaatimustenhallinta on vaatimusten tilan seuranta~~
- ~~Vaatimustenhallinta on sama kuin Requirements Management~~
- ~~Vaatimustenhallinta on osa Systems Engineeringiä~~

Vaatimustenhallinta on pohjimmiltaan ihmisten välistä viestintää, joka pyrkii yhteiseen ymmärrykseen

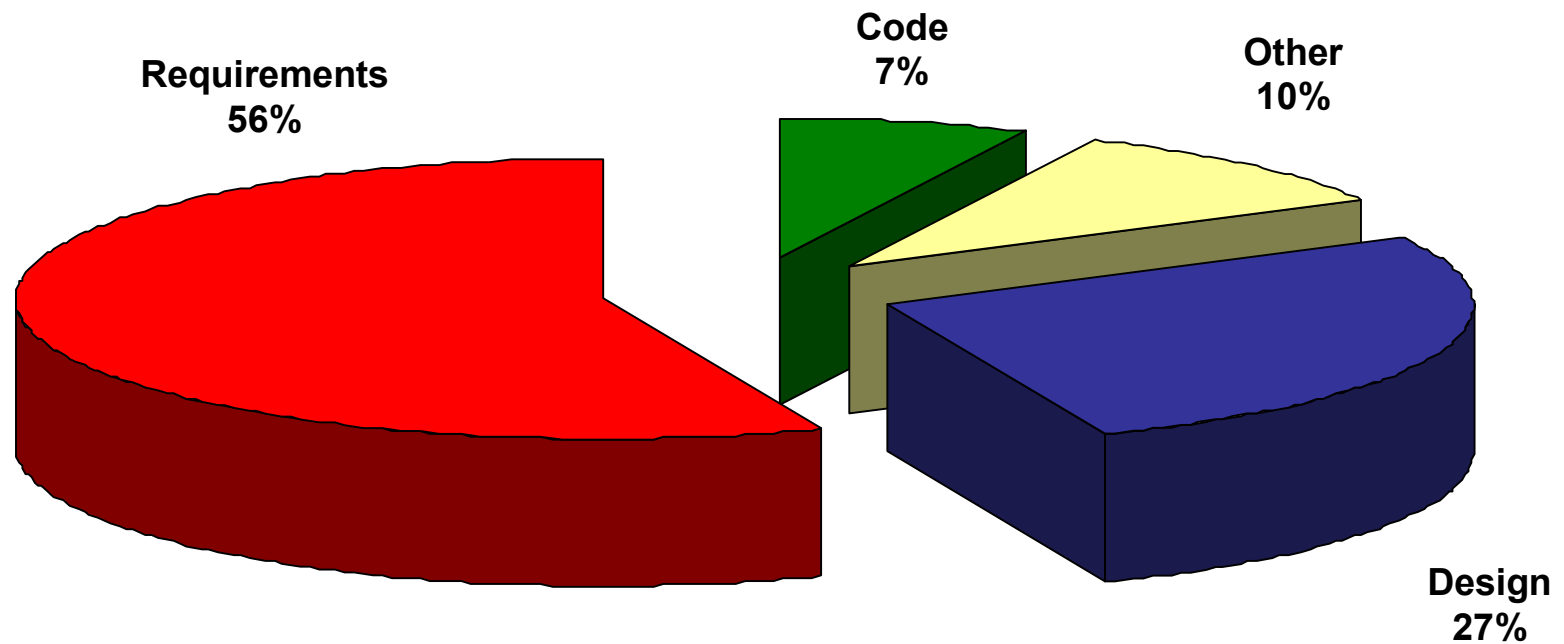
# Vaatimukset ja ISO 9001:2000

- Before submission of a tender, or the acceptance of a contract or order ... shall be reviewed by the supplier to ensure that:
  - the requirements are adequately defined and documented....
- The supplier shall establish and maintain documented procedures to control and verify the design of the product in order to ensure that the specified requirements are met.
- Design **input requirements** relating to the product, including applicable statutory and regulatory requirements, shall be identified, documented and their selection **reviewed** by the supplier for adequacy. Incomplete, ambiguous or conflicting requirements shall be resolved with those responsible for imposing these requirements.
- Design **output** shall be documented and expressed in terms that can be verified and validated against design input requirements.
- At appropriate stages of design, design **verification** shall be performed to ensure that the design stage output meets the design stage input requirements.

# Vaatimukset ja CMMI

- CMMI-versiossa 1.1. (SEI 2002) vaatimustenhallinta on ”key process area” tasolle 2 sisältäen seuraavat tavoitteet:
  - SG 1 Manage Requirements
    - SP 1.1 Obtain an Understanding of Requirements
    - SP 1.2 Obtain Commitment to Requirements
    - SP 1.3 Manage Requirements Changes
    - SP 1.4 Maintain Bidirectional Traceability of Requirements
    - SP 1.5 **Identify Inconsistencies between Project Work and Requirements**
  - GG 2 Institutionalize a Managed Process
    - (describes general goals for process improvement which need to be checked for requirements management processes, e.g. configuration management)

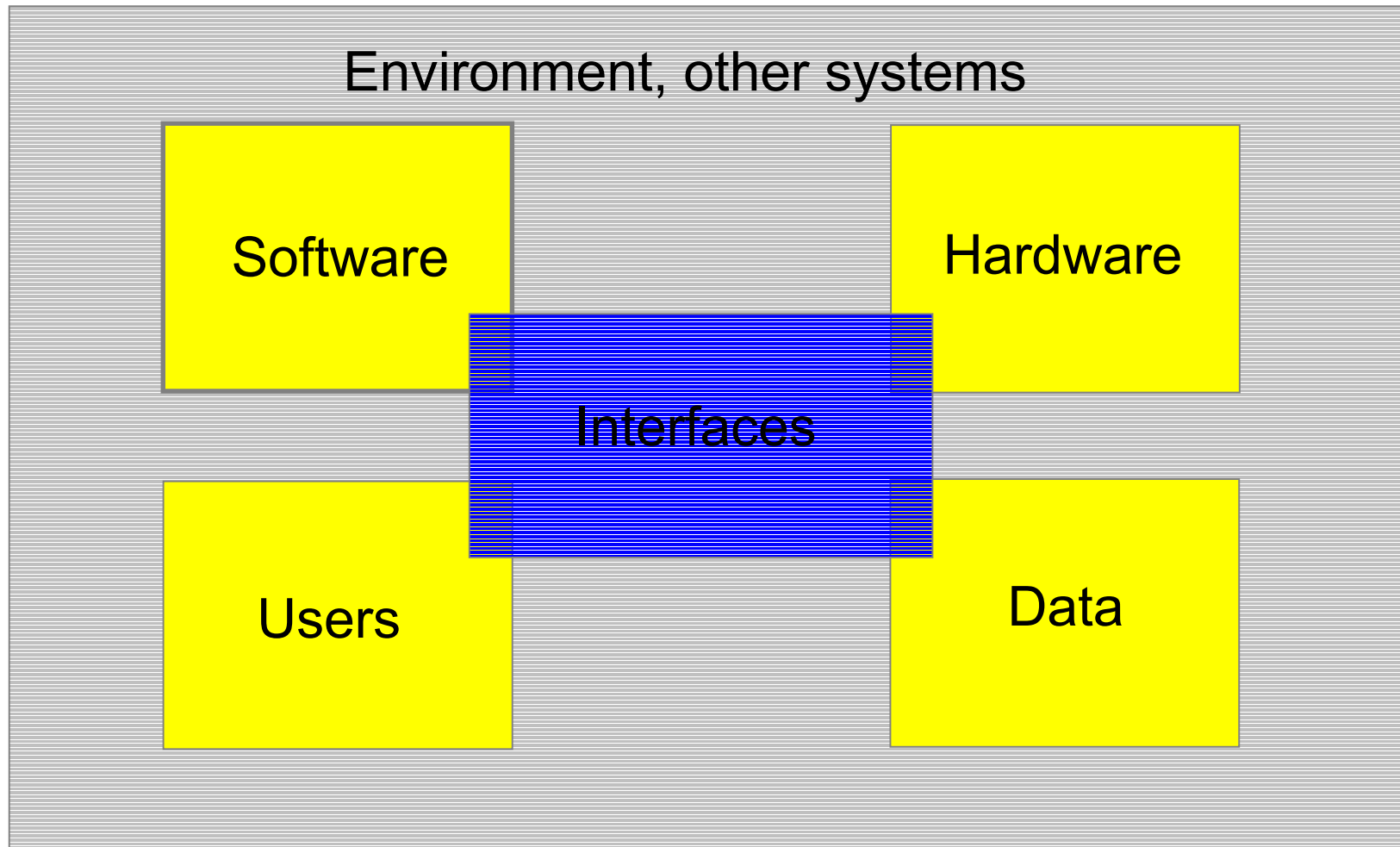
# Mistä virheet ja riskit ovat peräisin?



%-arvo kuvastaa syyn osuutta havaituista virheistä

Lähde: James Martin, An Information Systems Manifesto

# Ohjelmisto on vain osa kokonaissysteemiä



Ohjelmisto itsessään ei ole hyödyllinen kenellekään

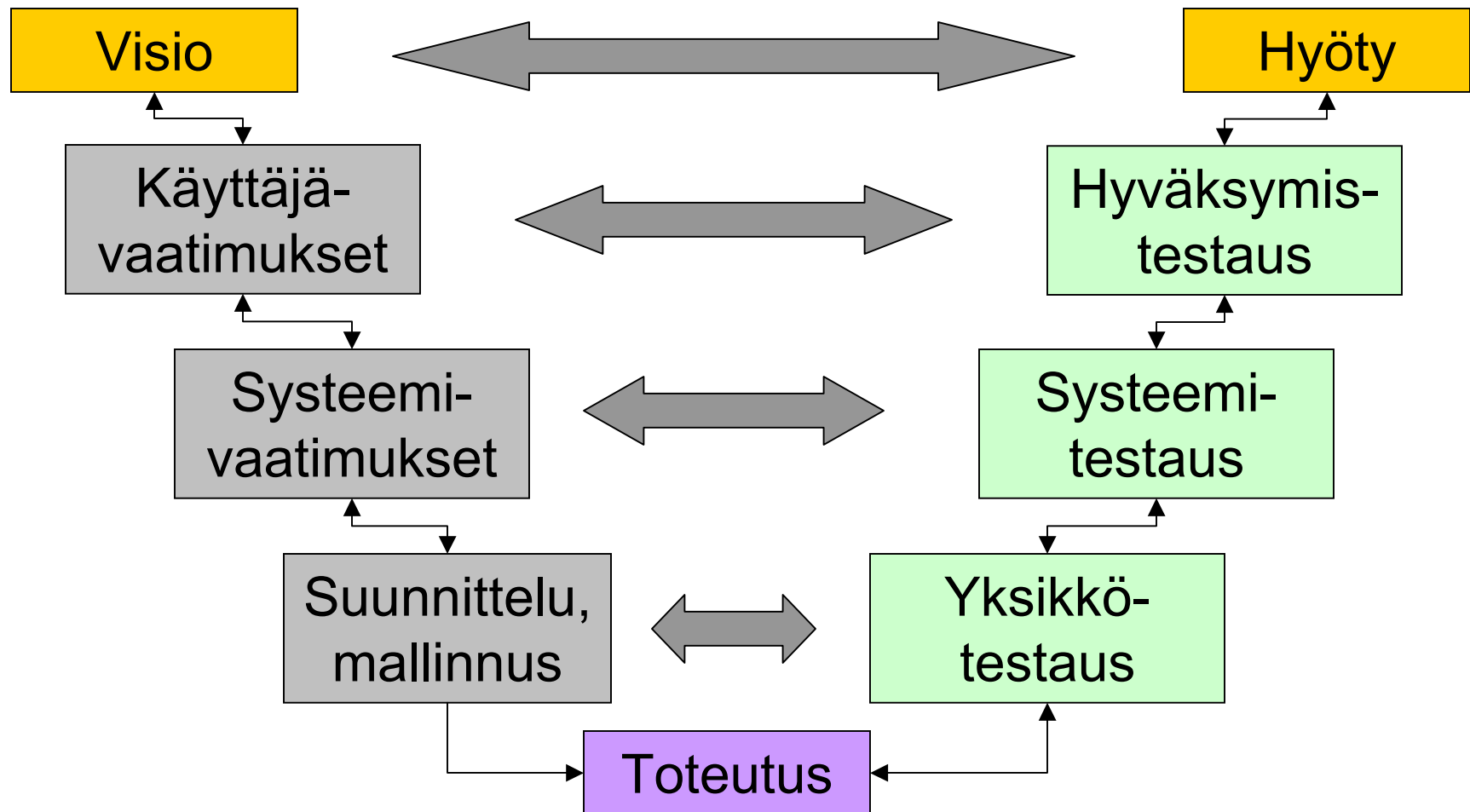
# Systemi ja ympäristö



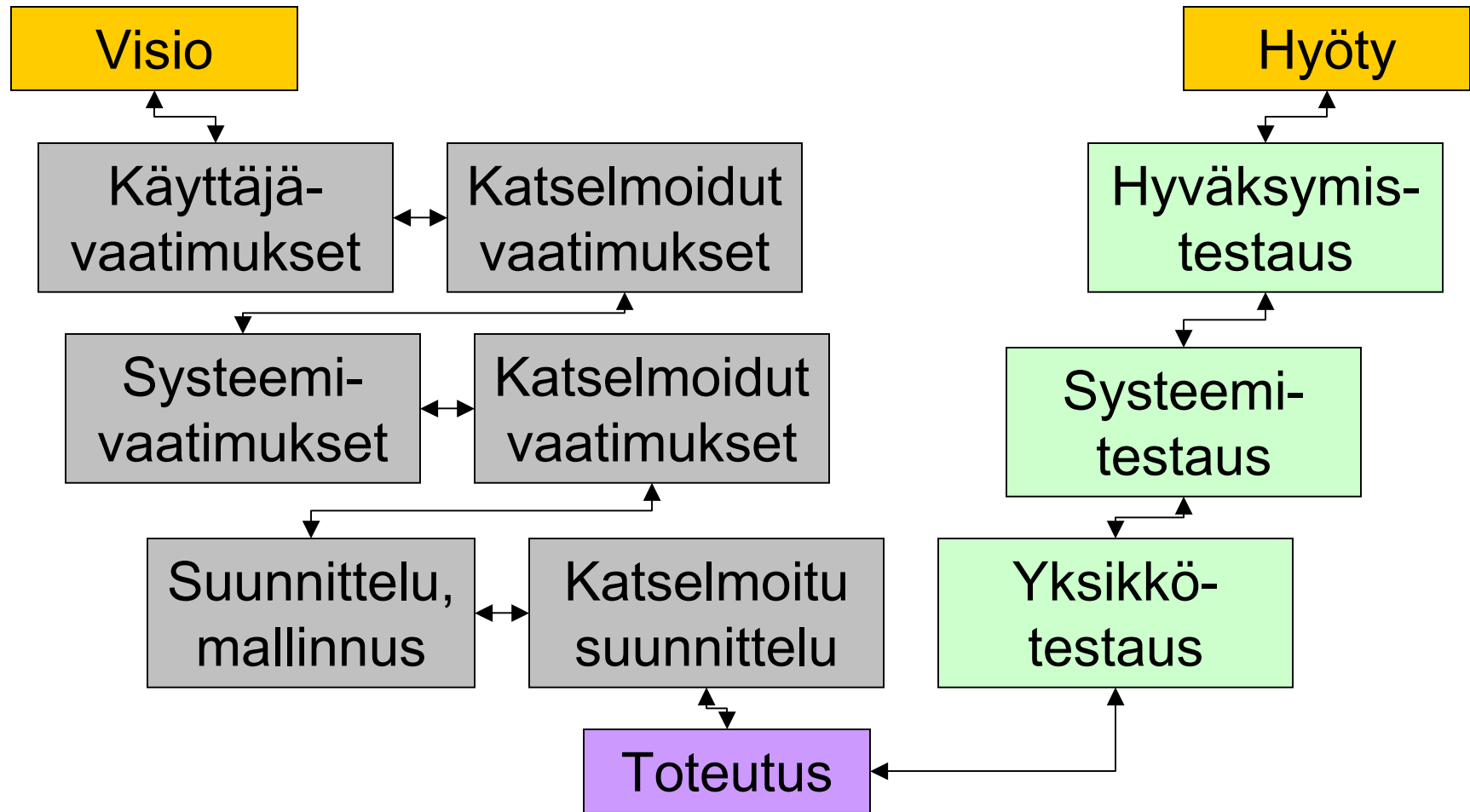
Helsingissä kapeat raiteet?  
Raiteet pohjattu betonin päälle?



# V-malli toteutuksesta

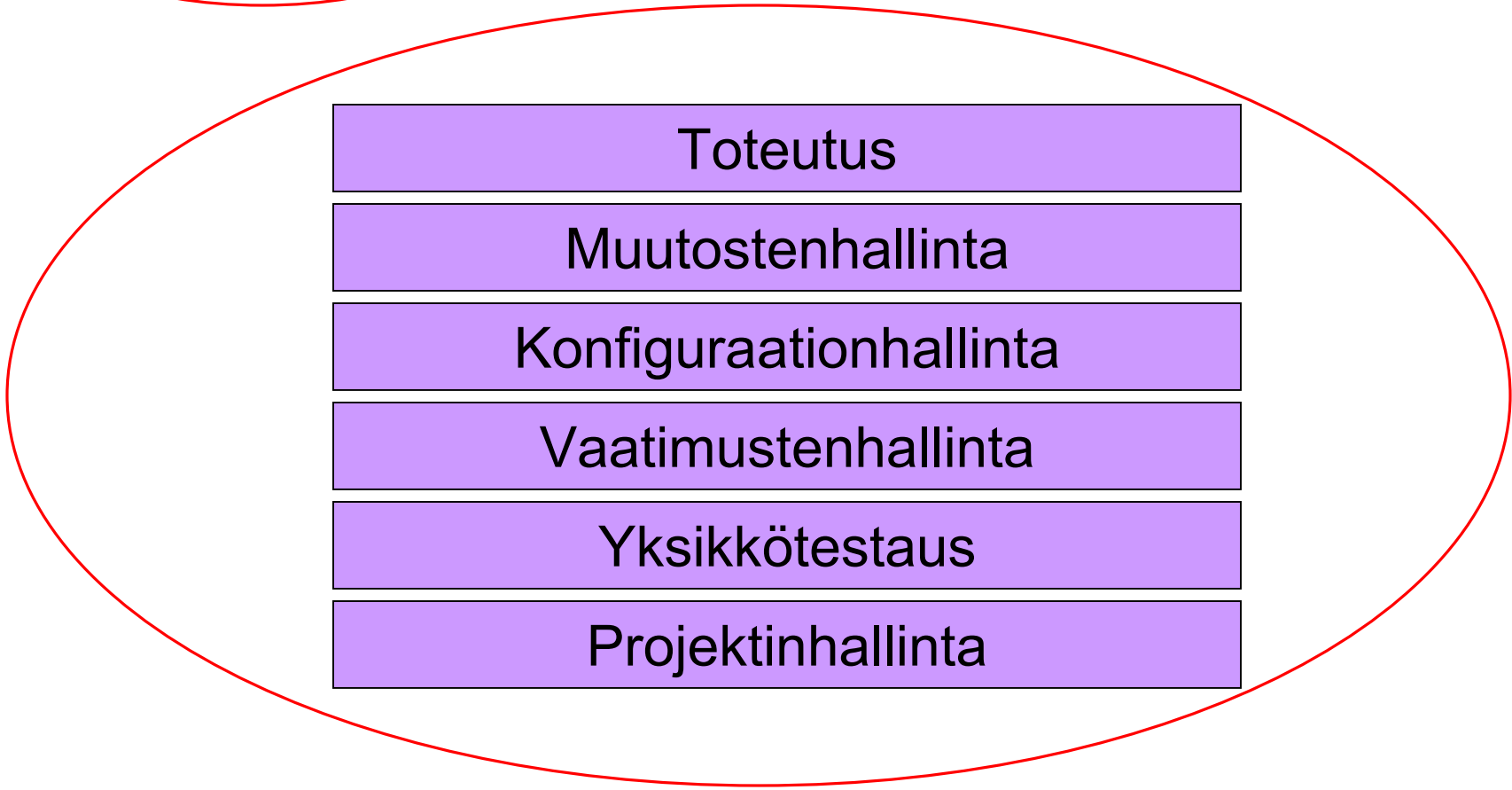
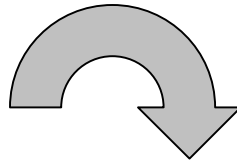


# Laajennettu V-malli toteutuksesta

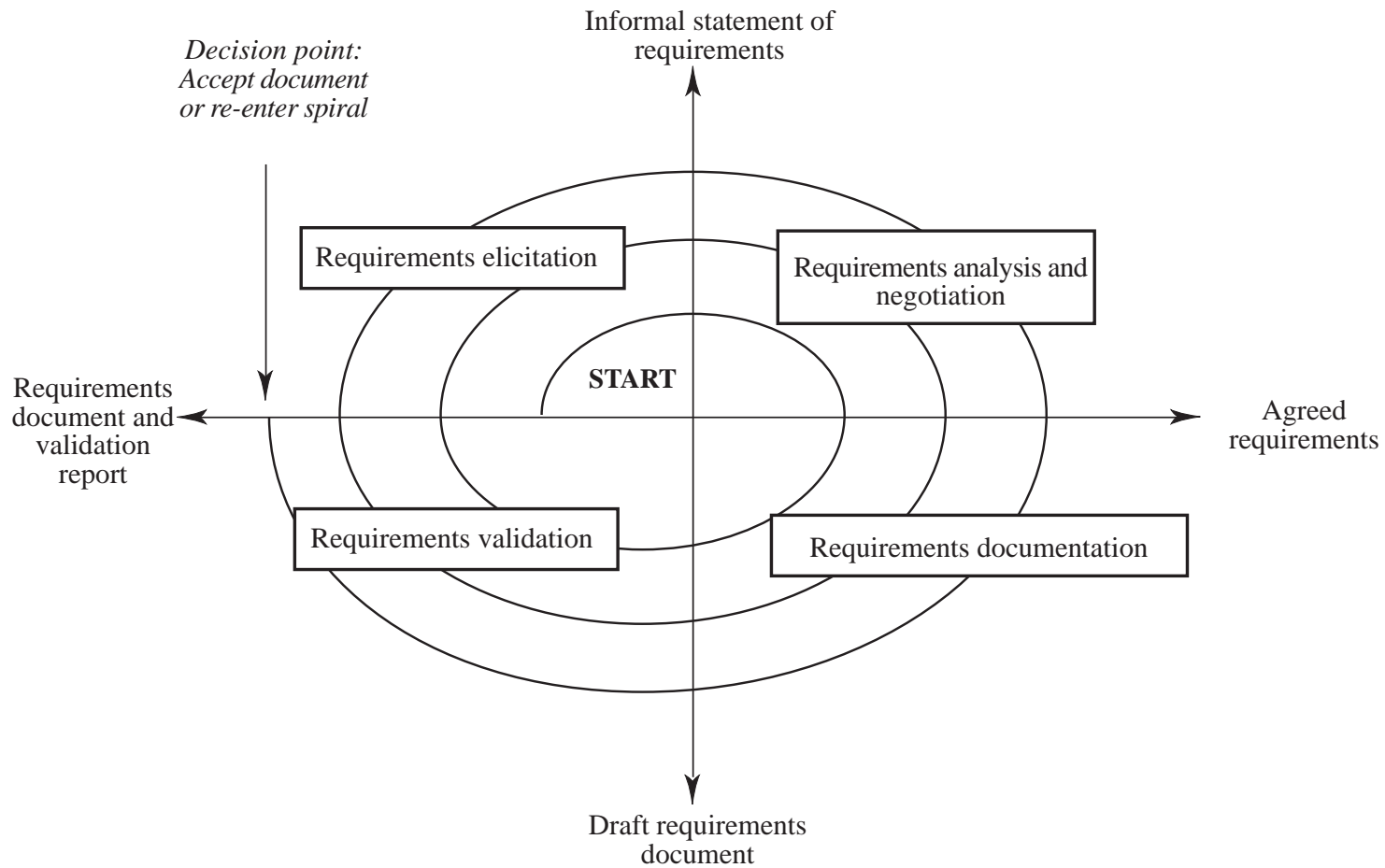


# Jatkuvat prosessit toteutuksessa

Toteutus



# Vaatimukset ovat aina iteratiivisia



# Vaatimusten kerääminen

- Määrittele tai käytä valmista rakenne pohjaa dokumentille
- Kirjoita vaatimukset talteen mahdollisimman aikaisin, vaikka ne ovat epätäydellisiä
- Tuota alustava dokumentti nopeasti, jotta saadaan mahdollisimman aikaisin palautetta
- Työstä vaatimuksia palautteen pohjalta, korjaamalla puutteita, virheitä ja ristiriitoja
- Pidä brainstorm- ja katselmointilaisuuksia (epämuodollisia) usein palautteen saamiseksi
- Palaute oikeilta käyttäjiltä on paljon parempi kuin asiantuntijoiden – käytä esim. prototyyppejä

Vaatimusten määrittelyprosessi on iteratiivinen!

# Hyvän vaatimuksen ominaisuudet

- Jokaisen yksittäisen vaatimuksen pitäisi olla:
  - **Oikea** – Teknisesti ja laillisesti mahdollista
  - **Täydellinen** – Ilmaista yksi ajatus tai toteama
  - **Selkeä** – Ymmärrettävä ja yksikäsitteinen
  - **Ristiriidaton** – Ei ristiriitoja muiden vaatimusten kanssa
  - **Testattavissa** – Toteutuminen voidaan todentaa
  - **Jäljitettävissä** – Tunnistettu yksikäsitteisesti, voidaan jäljittää
  - **Järkevä** – Toteutettavissa: kustannukset, aikataulu
  - **Riippumaton suunnittelusta** – Ei sido suunnittelua
  - **Tarpeellinen** – Tarvitaan tosiaan
  - **Priorisoitu** – Tarpeellisuus arvioitu ja prioriteetti asetettu

Kuinka tuotetaan tällaisia vaatimuksia?

# Hyvä vaatimusdokumentti

- Täydellinen – Tietoja ei puutu.
- Ristiriidaton – Ei ristiriitoja dokumentin sisäisesti tai muihin dokumentteihin.
- Muokattavissa – Dokumentti on muokattavissa ja tiedot muutoksista on ylläpidettävissä.
- Jäljitettävissä – Jokainen vaatimus dokumentissa pitäisi olla jäljitettävissä rakenteessa ylös- ja alaspäin.
- Luettava ja ymmärrettävä – Vaatimustenhallinta on viestintää.

Hyvä vaatimusdokumentti koostuu hyvistä vaatimuksista.

# Tekstin rakenne

## Before:

3.1 The XYZ system shall provide variance / comparative information that is timely, itemized in sufficient detail so that important individual variances are not obscured by other variances, pinpoints the source of each variance, and indicates the area of investigation that will maximize overall benefits.

Lähde: W.M.Wilson: Writing effective natural language requirements specifications

Crosstalk, February 1999 (<http://www.stsc.hill.af.mil/>)

## After:

### 3.1 Variance information

The XYZ system shall provide variance / comparative information.

#### 3.1.1 Information detail

Variance / comparative information shall be timely. Variance / comparative information shall be itemized in sufficient

##### 3.1.1.1 Level of details

Prevent important individual variances from being obscured by other variances.

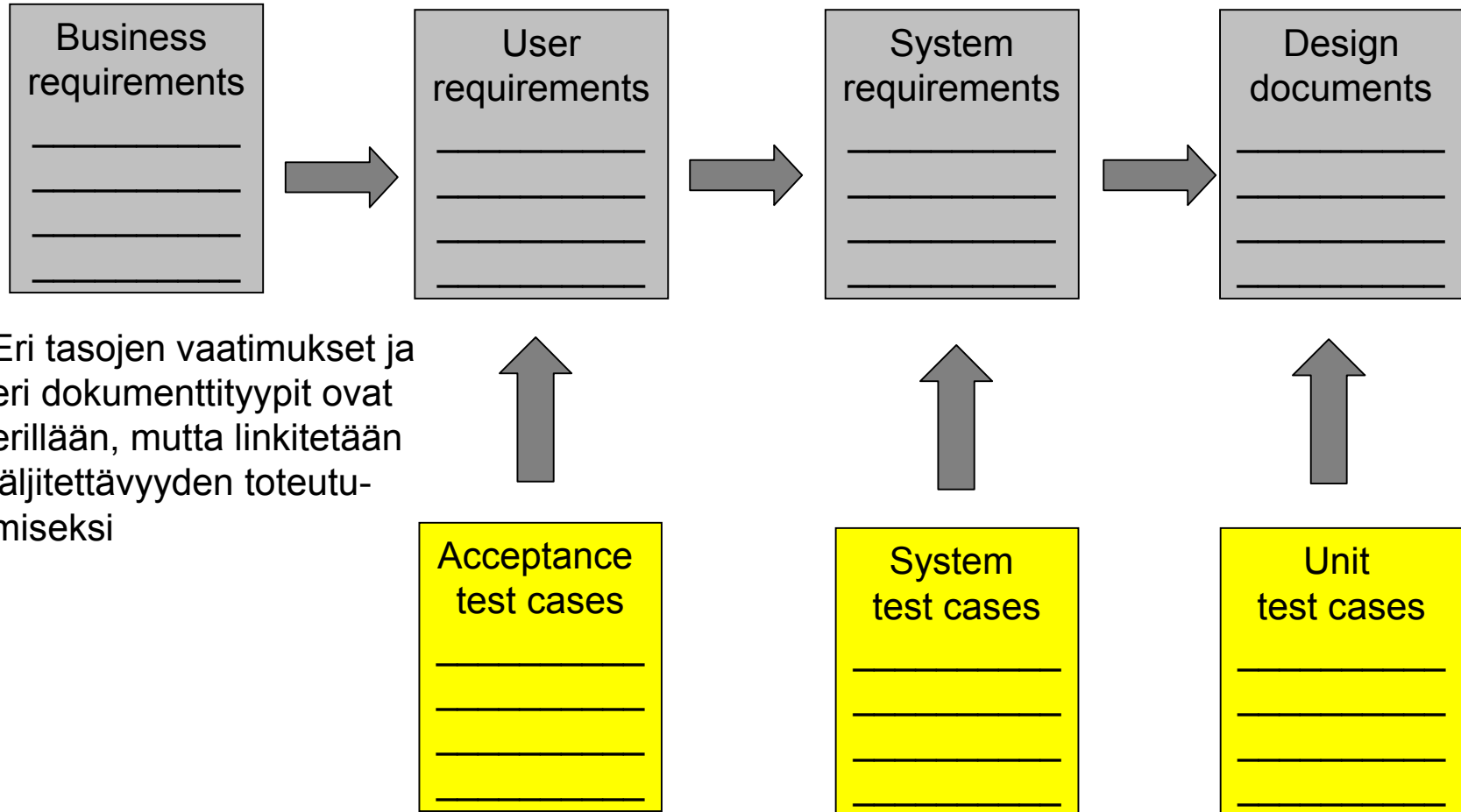
Pinpoint the source of each variance. ....



# Vaatimukset eivät ole koskaan täydellisiä

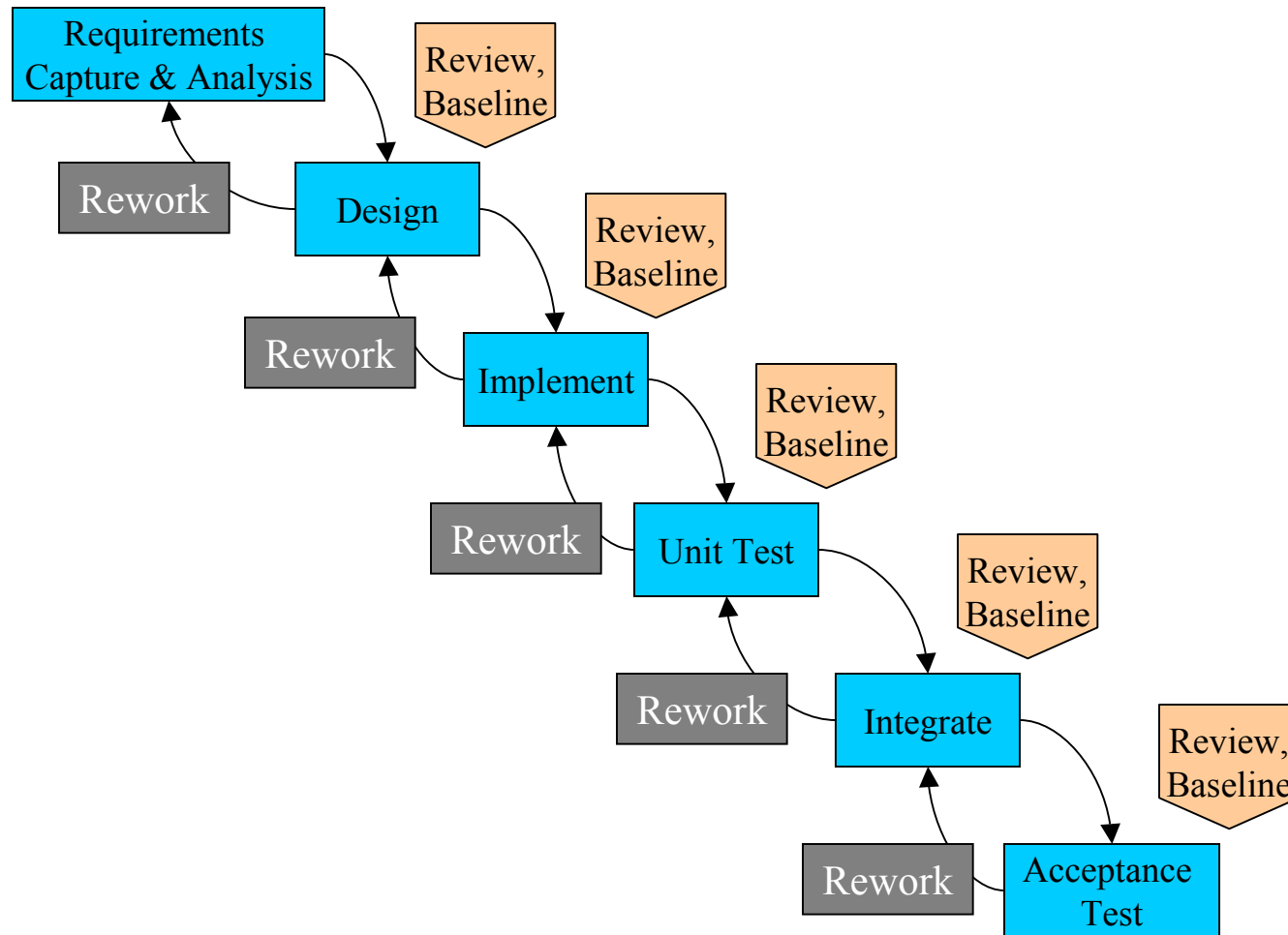
- Koska vaatimusten keruu ja hallintaprosessi on iteratiivinen, muuttuvat vaatimukset koko ajan.
- Muutoksia kertyy myös toteutuksen muista osioista, sekä testauksesta.
- Loppujen lopuksi lähtökohdaksi kannattaa ottaa, että vaatimukset muuttuvat koko kehityshankkeen elinkaaren ajan ja mahdollisesti ovat valmiit kun hanke loppuu.
- Täydellisiä vaatimuksia ei koskaan saada, ainoa mitä voi toivoa, on että vaatimukset lähestyvät täydellisyyttä.
- Pyrkimys lähestyä saavuttamatonta on kuitenkin hyödyllistä!

# Projektin dokumenttimalli

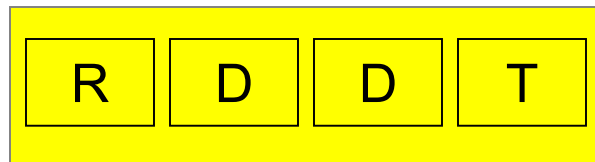


Eri tasojen vaatimukset ja eri dokumenttityypit ovat erillään, mutta linkitetään jäljitettävyyden toteuttamiseksi

# Mielikuvissa käytetään vesiputousmallia

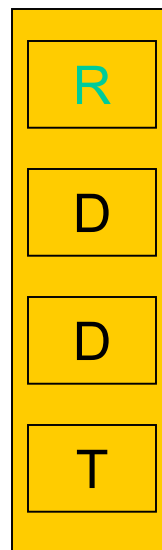


# Inkrementaalinen kehitys on myös "pop"



1st

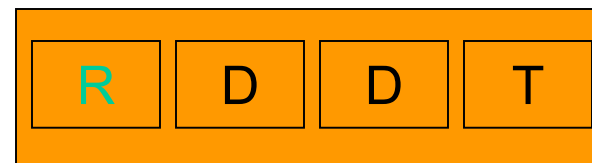
→ 1<sup>st</sup> increment taken into actual use?



← "prototype"

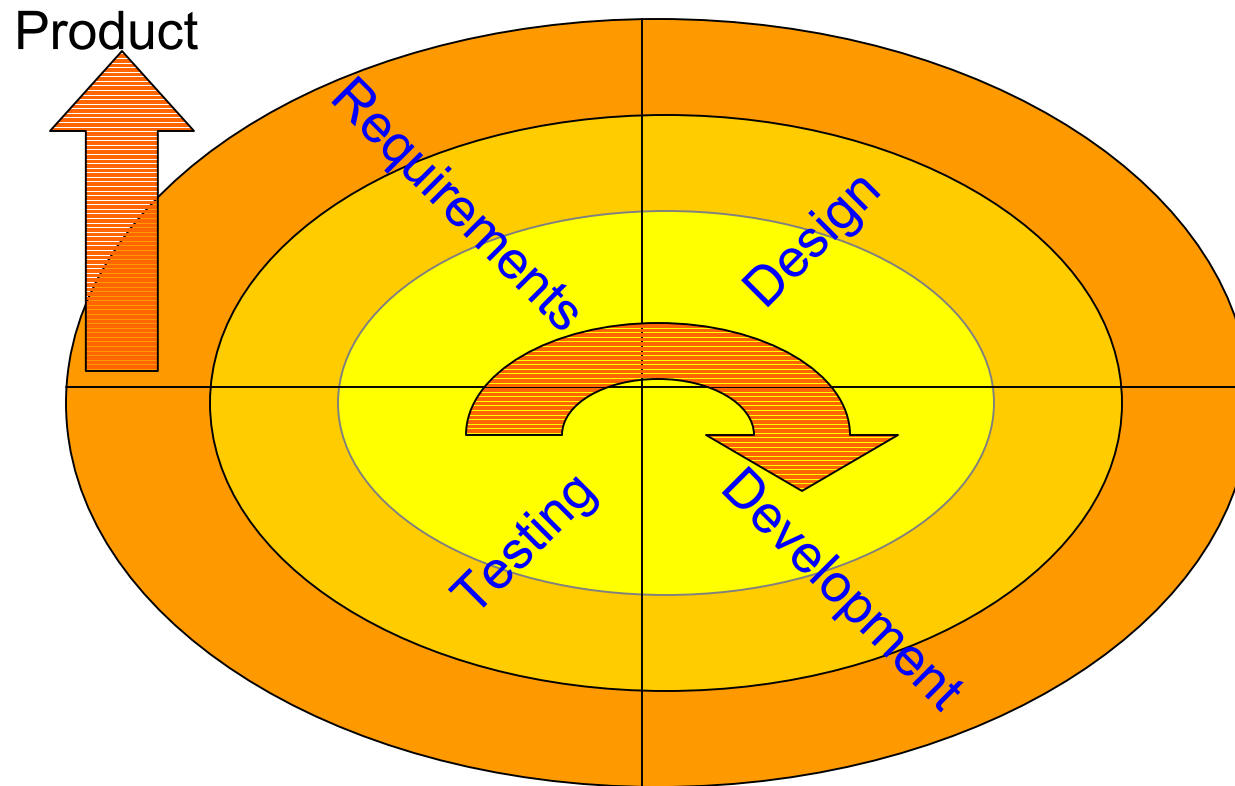
Kuinka vaatimukset kerätään ja hallinoidaan? Kuinka vaatimusten ja niiden muutokset arvioidaan vaikutuksina suunnitteluun, toteutukseen sekä testaukseen?

2nd



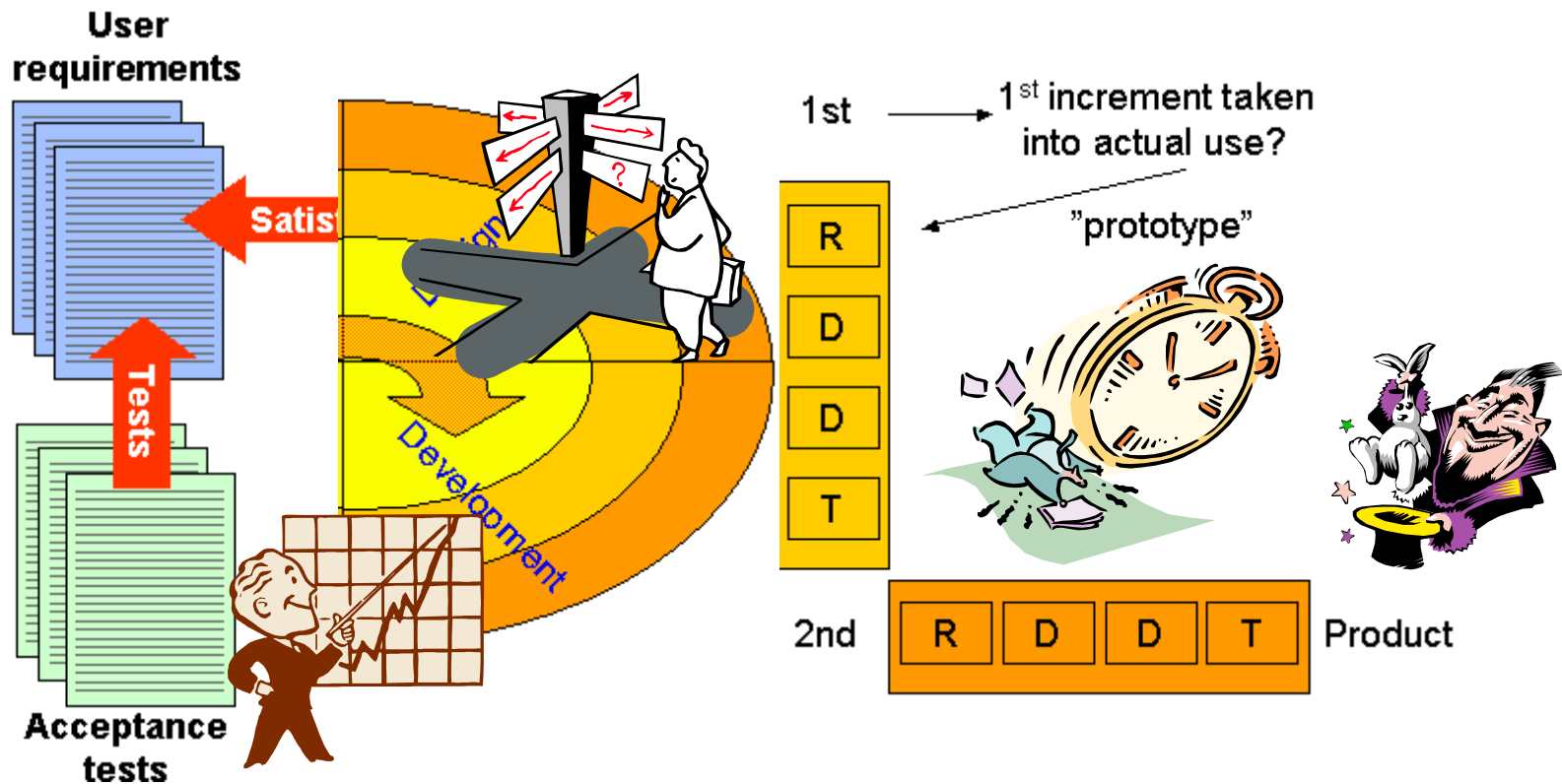
Product

# Inkrementaalinen on vain usein iteratiivista



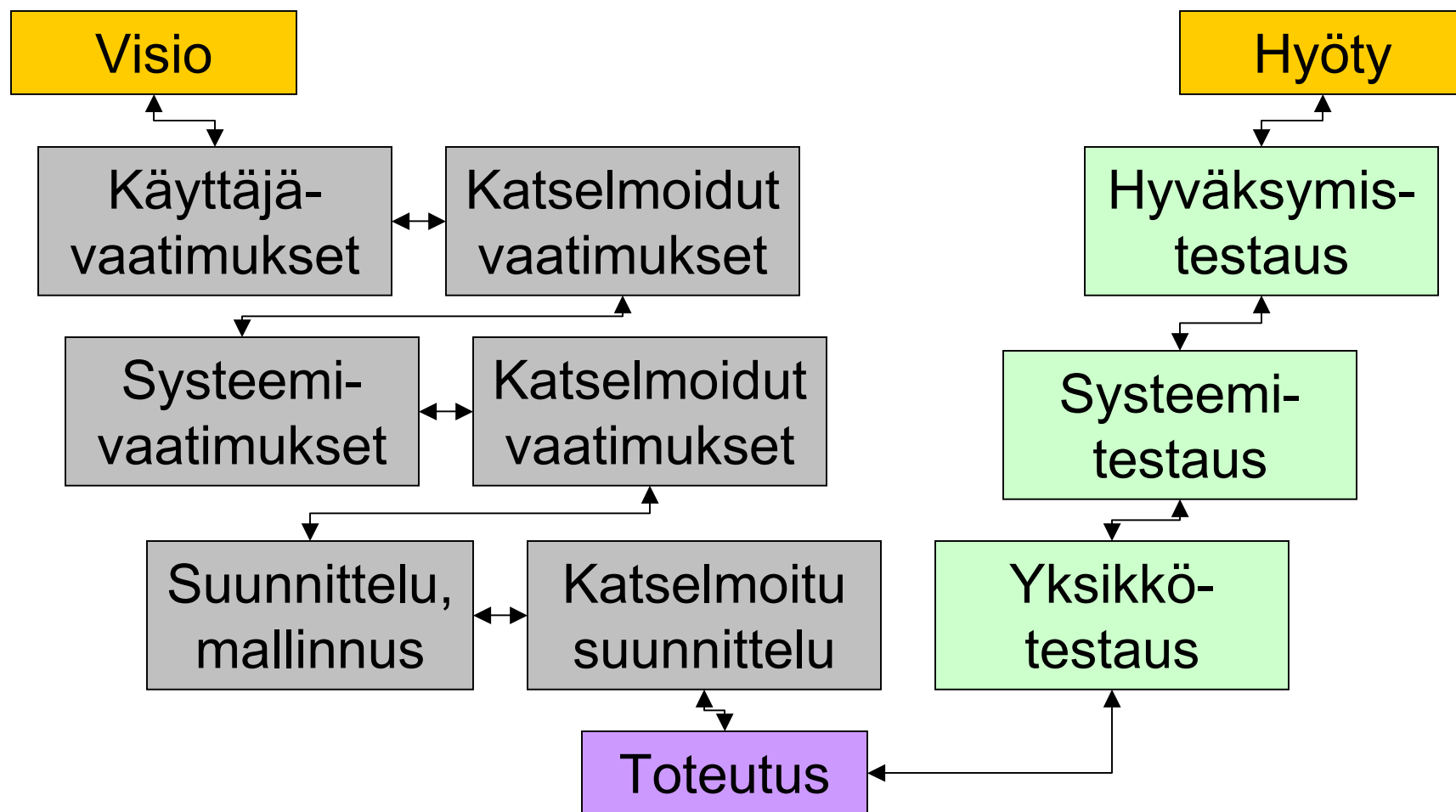
Kuinka vaatimukset kerätään ja hallinnoidaan? Kuinka vaatimusten ja niiden muutokset arvioidaan vaikutuksina suunnitteluun, toteutukseen sekä testaukseen?

# Pitääkö myöntää, että kaikki on hässäkkää?

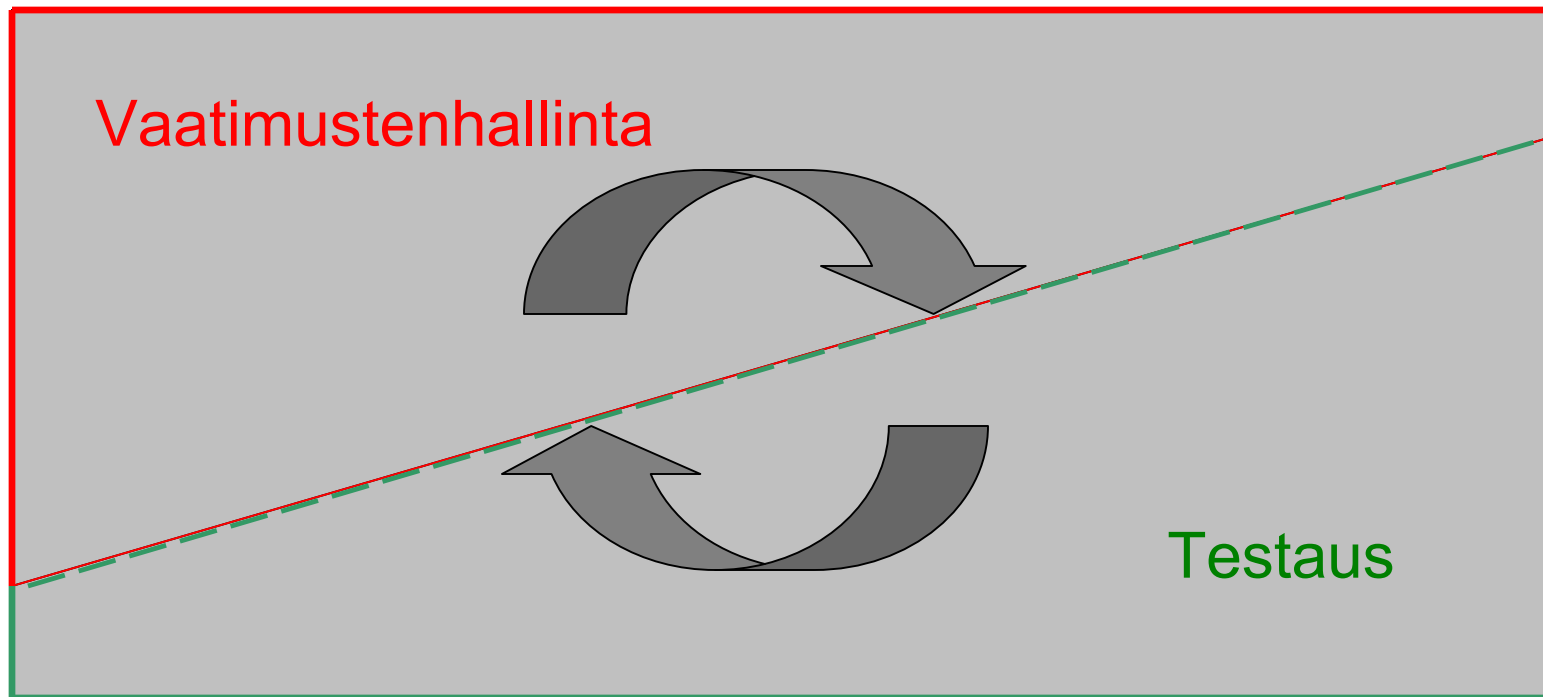


Käytännön hankkeissa prosessimalleja usein yhdistellään tai ne yhdistyvät, suunnitelmallisesti tai suunnitelmatta.

# Jos ajatellaan V-mallia dokumenttimallina



# Tiivistetty "prosessi" ajan kuvana

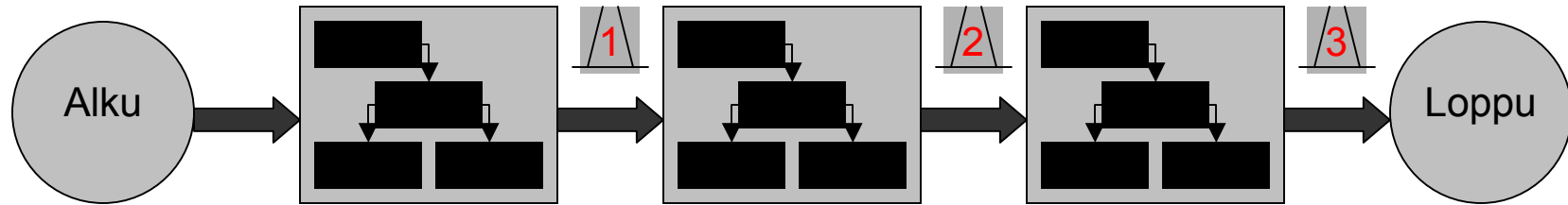


Jos otetaan kirjallisesti laatustandardien näkemys toteutushankkeista, niin niissä on kysymys vaatimusten tilan seurannasta sekä vaatimusten toteutumisen testaamisesta.

Aika



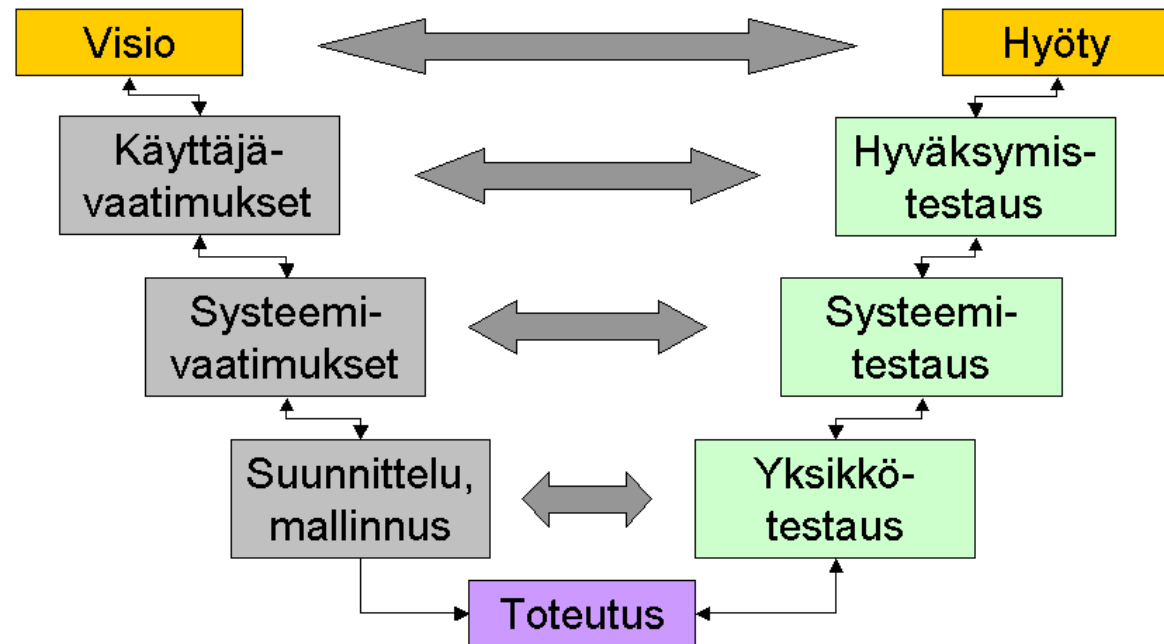
# Milestone –pisteet kiinnittävät prosessin



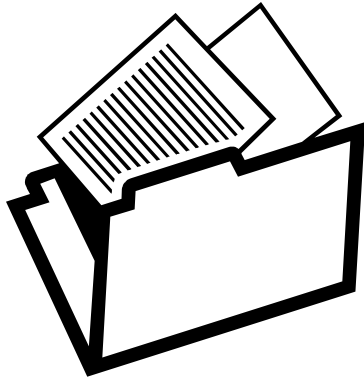
- Milestone –pisteet 1,2 ja 3 ovat ne kohdat prosessia jossa:
  - On selkeästi määritelty mitä pitää olla tehtynä jotta pystytään jatkamaan eteenpäin seuraavaan vaiheeseen
  - Voidaan tarkistaa myös tehdyn työn yhteensopivuus projektivision tai liiketoiminnallisten tavoitteiden kanssa
  - Tehdään mahdollisesti go / no-go –päätös hankkeen jatkamisesta.
- Esim: milestone 1:ssä pitää olla asiakasvaatimukset koottuna sekä katselmoituina. Asiakasvaatimukset pitää olla linkitettynä alustaviin systeemivaatimukseen, sekä hyväksymistestitapauksiin. Muutostenhallinta käynnistetään tämän jälkeen.

# Vaatimusten kytkeminen testaukseen

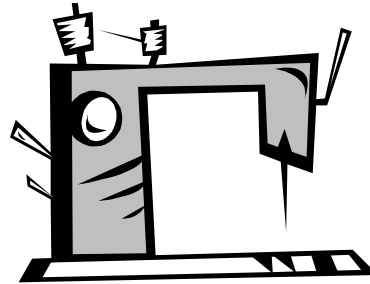
- Jos hyvälaatuiset vaatimukset todetaan toteutetuiksi ja testatuiksi hyväksytysti, niin eikö silloin ole kaikki tehty?



# Muutosten ongelma



Tehdään  
"määrittely",  
"speksit" tms.



Joiden perusteella  
tehdään tuote



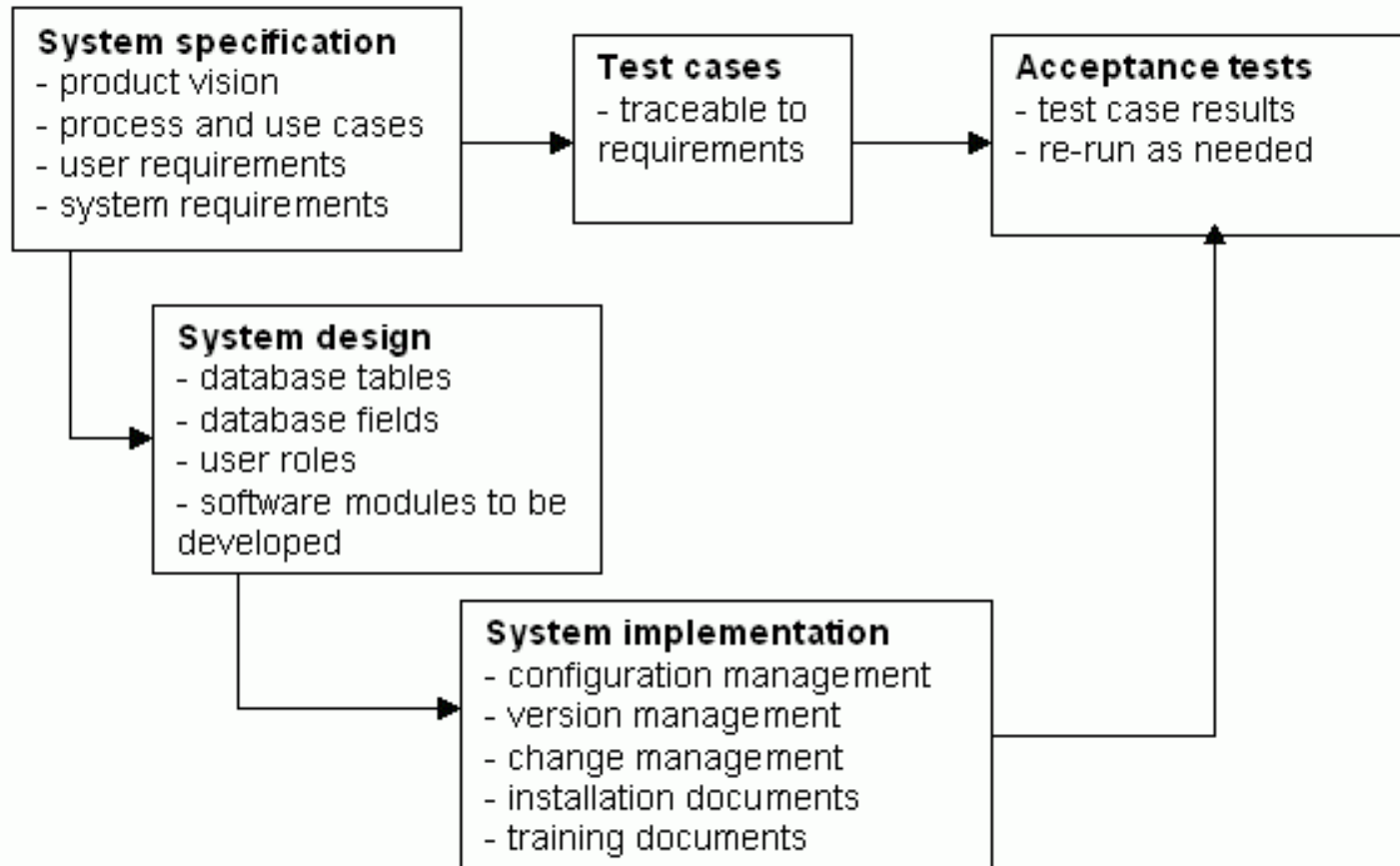
Muutokset kohdistuvat tyypillisesti vain toteutukseen ja sen dokumentaatioon (esim. ohjelmiston lähdekoodiin), eikä määrittelyyn. Täten lopputuloksena on tuote joka ei enää vastaa määrittelyä, koska tuote ja määrittely ovat muutosten johdosta eronneet toisistaan.

Samoin jos toteutusta vain muutetaan, ei testaus koskaan testaa oikeita asioita.

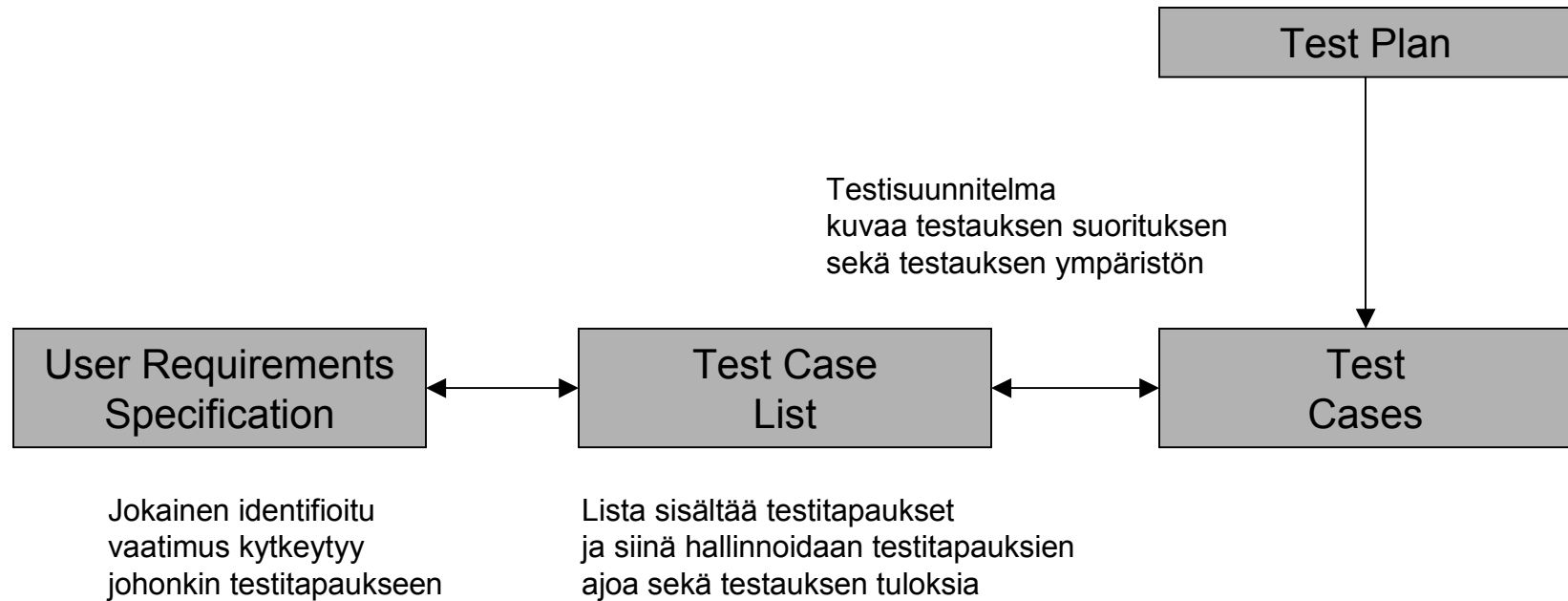
# Muutostenhallinta on tärkeää

- Vaatimusten, suunnittelun, toteutuksen sekä testauksen välinen jäljitettävyyden on ehdoton, jotta voidaan hallita muutoksia sekä arvioida niiden aiheuttamia kustannuksia.
- Jäljitettävyyden kautta voidaan käydä lävitse muutoskohtaan liitetyt muut vaatimukset, suunnittelun sekä testauksen ja arvioida niiden muutostarpeet.
- Muutostenhallinnan prosessin pitää toteuttaa seuraavia vaiheita:
  - muutostarpeet esitetään dokumentoidusti muutosehdotuksina
  - muutosehdotukset pitää arvioida niiden tarpeellisuuden, kustannuksien ja aikatauluvaikutuksien osalta
  - pitää tehdä päätös muutosehdotuksen toteuttamisesta tai hylkäämisestä
  - jos muutos päätetään toteuttaa, pitää tämä tieto saada viestittyä toteutusta tekeville henkilöille
  - muutos pitää dokumentoida
  - ja lopuksi päätetty muutos pitää myös toteuttaa ja varmistaa toteutus.

# Esimerkki dokumenttimallista



# Käytännön toteutus dokumenttimallista



# Menetelmiä testata vaatimusdokumentteja

- Tarkastus, siis tekstin läpikäynti oikeakielisyyden tarkistamiseksi.
- Katselmointi, eli dokumenttien läpikäynti sisällön oikeellisuuden ja täydellisyyden tarkistamiseksi.
- Suunnittelun ja testauksen käynnistäminen vaatimusten pohjalta tuo esille puutteita vaatimusdokumenteissa.
- Tarkistuslistat vaatimusten oikeellisuudesta.
- Käyttäjätyyppien läpikäynti – onko kaikilta käyttäjiltä vaatimuksia?

Kriteeri		Kpl vaatimuksista		Painoarvo	Arvosana
		OK	Hylätty		
1	Oikeellisuus	87	13	1,00	87
2	Kattavuus	90	10	0,50	45
3	Selkeys	67	33	1,50	100,5
4	Ristiriidattomuus	93	7	0,80	74,4
5	Yksikäsitteisyys	70	30	1,00	70
6	Jäljitettävyys	48	52	1,00	48
7	Liityntä ylempään vaatimukseen	78	22	1,50	117
8	Testattavuus	30	70	1,00	30
9	Toteutusriippumattomuus	84	16	0,50	42
10	Toteutettavuus	97	3	1,20	116,4
<b>Yhteensä</b>		<b>744</b>	<b>256</b>		
<b>Arvosana (asteikolla 1-10)</b>					<b>7,3</b>

Taulukon lähde: Kosola / Pasivirta: Vaatimustenhallinnan soveltaminen puolustusvoimissa

# Requirements-based Testing

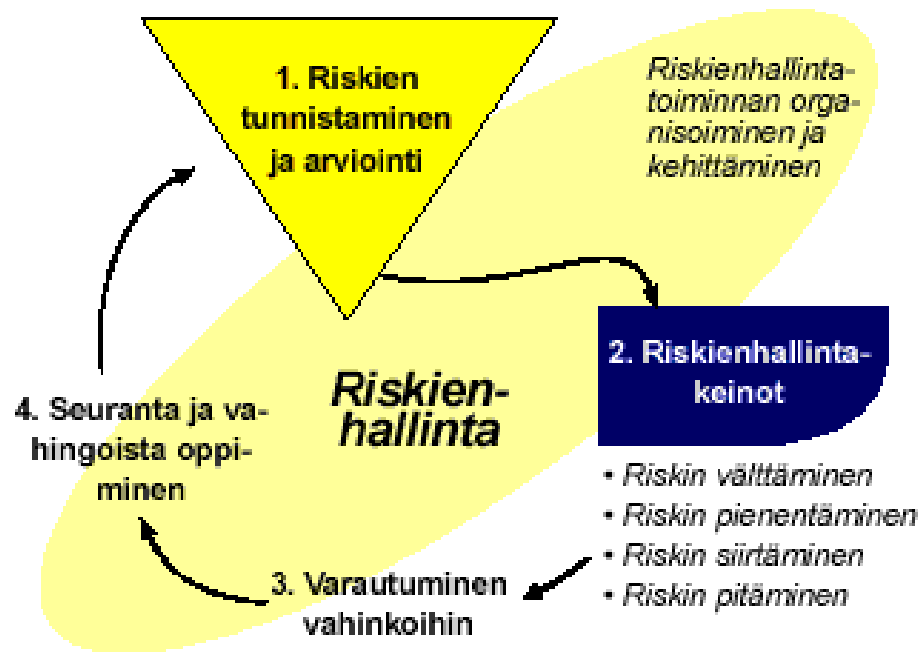
- RBT on on Bender RBT:n määrittelemä prosessimalli vaatimusohjatuille ohjelmistotestaukselle
- Lyhyesti:
  - Menetelmässä varmistetaan, että vaatimukset ovat muodollisesti ja sisällöllisesti oikein katselmoimalla
  - Katselmoidut vaatimukset mallinnetaan syy-seurauskaaviossa
  - Testitapaukset luodaan syy-seurauskaavioiden pohjalta
  - Testitapaukset katselmoidaan
  - Käytetään testitapauksia suunnittelun ja toteutuksen katselmoinnissa
  - Testataan tuote testitapauksilla.



# Testaussuunnittelu testaa myös vaatimuksia

- Kun vaatimuksista lähdetään jo varhaisessa vaiheessa tekemään testitapauksia, tulee saman tien selvitettyä:
  - ovatko vaatimukset selkeitä
  - voiko niiden perusteella tehdä testausta.
- Välttämättä ei tarvitse tuottaa laajoja testitapauksia, vaan enemmänkin miettiä minitestitapaustasolla vaatimusta:
  - Esimerkki pankkiautomaatista: ”Käyttäjän pitää olla mahdollista nostaa tililtään kerrallaan maksimissaan 400 euroa tai tilin saldon verran”.
  - Testivaatimuksia:
    - Nosta tililtä 600 euroa, jos tilillä on niin paljon
    - Nosta tililtä 158,67 euroa
    - Nosta tililtä kymmenen kertaa saman päivän päivän aikana 400 euroa
  - Minitestitapauksia voidaan tuottaa muodollisemmin riskianalyysin kautta.
- Näitä minitestitapauksia ei välttämättä käytetä lopullisessa testauksessa, koska sinne saavuttaessa testitapauksetkin ovat voineet muuttua.

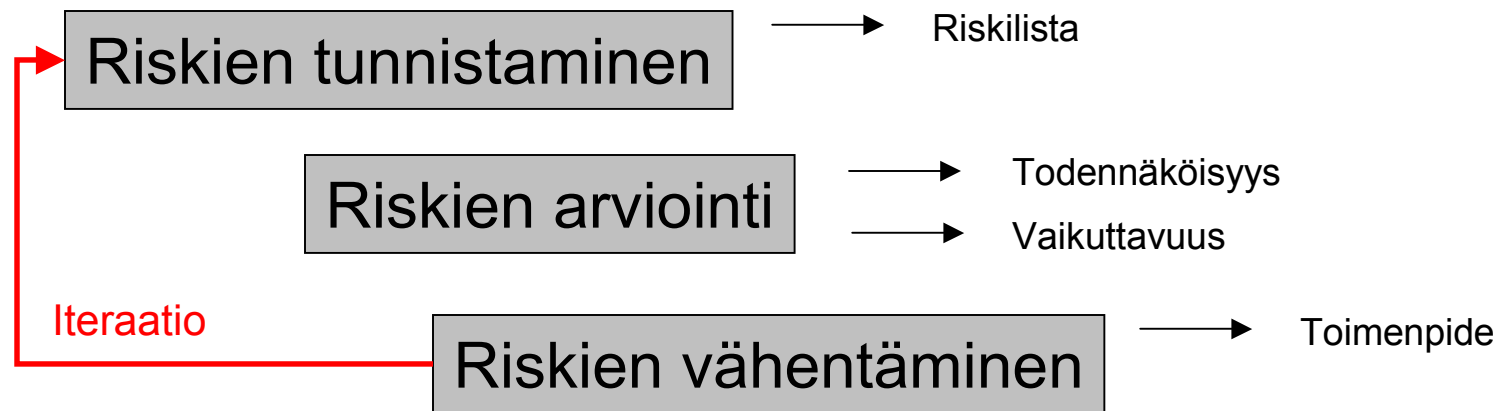
# Riskienhallinta



Riski on mahdollisuus, että haitallinen tapahtuma toteutuu. Tässä määritelmässä riskillä viitataan tilanteeseen, jossa on mahdollista, mutta ei täysin varmaa, että esiintyy ei-toivottu tapahtuma, jolla on haitallisia seurauksia. Siten määritelmään sisältyvät niin todennäköisyys kuin seurauksetkin.

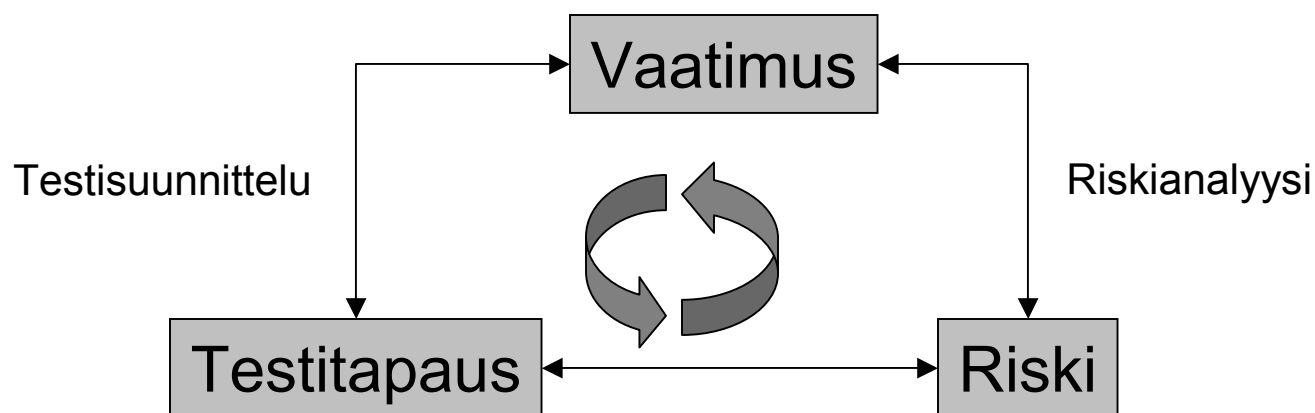
Lähde: <http://www.pk-rh.com/>

# Riskienhallinnan prosessi



Code	Risk	Prob.	Imp.	P*I	Action	Output
P1	The schedule is too tight	6	5	30	The schedule and project progress is followed up weekly by project manager and in every second week by the project steering group	Project's status in progress reports. Decided actions made by steering group e.g. for giving extra time, if needed.
T2	It is hard to integrate the product to customer's environment or the integration fails	4	5	20	The integration to the customer's environment is tested a couple of times before the final product version.	Integration results that can be used when planning the last integration.
P10	The product support will be inadequate	4	5	20	The support will be planned during the project as a separate track.	Support plan, Support agreement

# Vaatimukset, riskit ja testaus

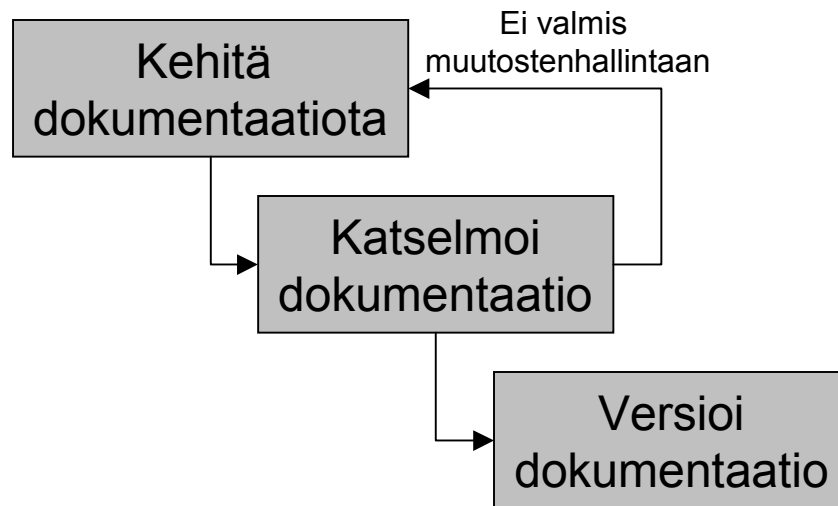


Vaatimukset täsmentyvät riskianalyysin sekä testaus suunnittelun kautta, jokainen tässä osiossa tuottaa lisää aineistoa toisiin:

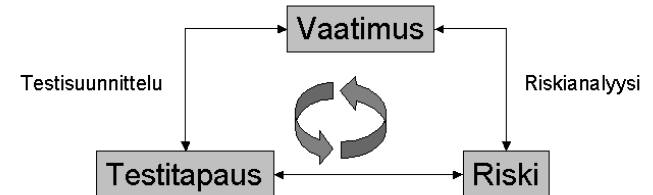
- Vaatimuksista voidaan lähteä hahmottamaan riskejä
- Riskit tuottavat lisää vaatimuksia sekä testitapauksia
- Testitapaukset pakottavat selventämään vaatimuksia

Kaikki tämä muutos ja kehitys tapahtuu iteratiivisesti.

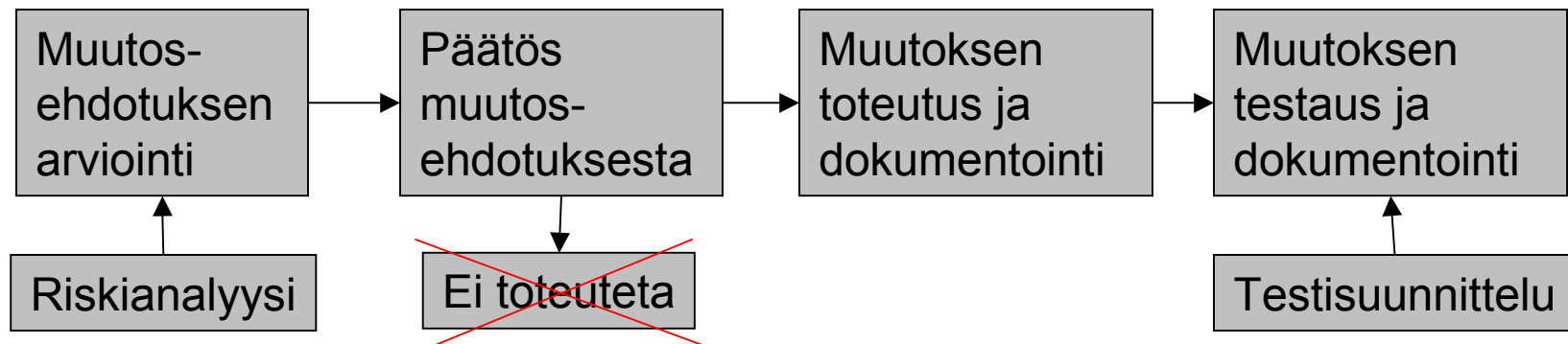
# Korkean tason malli kehitysprosessista



## Dokumentaation kehitysvaihe



## Dokumentaatio muutostenhallinnassa



# Hyviä käytäntöjä vaatimustenhallinnassa

<b>Best practice</b>	<b>Cost of introduction</b>	<b>Cost of application</b>
Prioritize requirements	Low	Low to moderate
Involve customers and users throughout RE	Low	Moderate
Use peer-reviews, scenarios etc. to verify and validate requirements	Low	Moderate
Allocate 15 to 30 percent of total project effort to RE	Low	Moderate to high
Identify and consult all likely sources of requirements	Low to moderate	Moderate
Provide specification templates and examples	Low to moderate	Low
Develop complementary models together with prototypes	Low to moderate	Moderate
Maintain a traceability matrix	Moderate	Moderate
Assign skilled project managers and team members to RE activities	Moderate to high	Moderate

Taulukon lähde: Hofmann - Lehner: Requirements engineering as a success factor in software projects  
IEEE Software, July/August 2001

# Lopussa kiitos seisoo

- Jos kehitys on iteratiivista ja muutoksia hoidetaan systemaattisen muutostenhallinnan kautta, voidaan todeta, että hankkeen lopussa dokumentaatio vastaa sitä mitä tehtiin.
- Onko lopputulos taas sitä, mitä alunpitäen alettiin tekemään...
- Joka tapauksessa, dokumentaatio – vaatimusspeksit, suunnitteludokumentaatio, testaussuunnitelmat, testitapaukset ovat nekin valmiit vasta kun projekti on loppu ja tuote / palvelu käytössä.
- Hässäkkä, mutta suunnitelmallinen hässäkkä.

# Testaajilla on kuitenkin laajempi hässäkkä

## Vaatimukset vs. Testaus- näkökulma vaatimuksiin

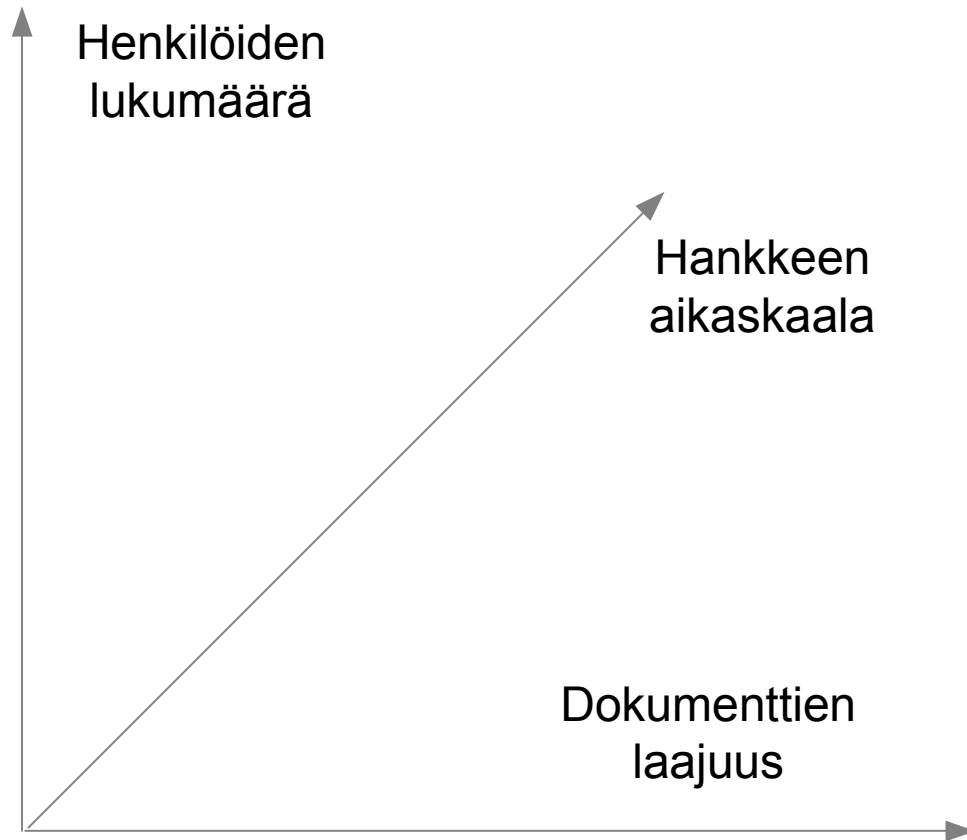


2.5.2005 11

Lähde: Maaret Pyhäjärvi: Ilman vaatimuksiakin voi testata



# Milloin tarvitaan työkalutukea?



Kokonaisuuden hallinta voi edellyttää työkalutukea, jotta dokumentaatio saadaan ylläpidettyä ja varsinkin dokumenttien välinen jäljitettävyyden ylläpidettyä.

Pienissä projekteissa dokumentaatiota voidaan ylläpitää ns. office-ohjelmistoissa, mutta henkilöiden lisääntyminen, pitkä aikaskaala sekä dokumentaation laajuus kasvattaa tarvetta prosesseja tukevan ohjelmiston käyttöön.

# Hyvää lukemista

- Robertson: An Early Start to Testing: How to Test Requirements  
<http://www.systemsguild.com/GuildSite/SQR/Testreqs.html>
- Gerrard: Testing Requirements  
<http://www.evolutif.co.uk/TestReqs/TESTREQS.html>
- Wiegers: Writing Quality Requirements  
<http://www.processimpact.com/articles/qualreqs.html>
- Bach: Risks and Requirements-based testing  
[http://www.satisfice.com/articles/requirements\\_based\\_testing.pdf](http://www.satisfice.com/articles/requirements_based_testing.pdf)
- Bender BRT Inc: Bender Requirements Based Testing  
<http://www.benderrbt.com/postionpap.htm>
- Boehm: Anchoring the Software Process  
<http://sunset.usc.edu/publications/TECHRPTS/1995/usccse95-507/ASP.html>
- Vaatimusten testattavuus –tilaisuus (Sytyke & Laatuokeskus)  
<http://pcuf.fi/sytyke/kerhot/testaus/historia.htm>
- Kosola / Pasivirta: Vaatimustenhallinnan soveltaminen puolustusvoimissa  
[http://www.mil.fi/paaesikunta/paaesikunta/sotatalous/raportit\\_julkaisut/Vaatimustenhallinnan\\_soveltaminen\\_pvssa.pdf](http://www.mil.fi/paaesikunta/paaesikunta/sotatalous/raportit_julkaisut/Vaatimustenhallinnan_soveltaminen_pvssa.pdf)

# Kiitos mielenkiinnosta!

Pekka Mäkinen, konsultti, yrittäjä

Sähköposti      Pekka.Makinen@SoftQA.fi

Yrityksen kotisivu      <http://www.softqa.fi/>

Soft  
QA