



# No test levels needed in agile software development!

Leo van der Aalst

Lector Software Quality and Testing

PNSQC, October 11h 2011

Portland

- About agile
- Testing in agile software development
- No test levels needed
- Product risk analysis (one-table approach)
- Summary

- **Rapid Application Development (RAD)**
  - 80's, Barry Boehm, Scott Shultz and James Martin
- **Scrum**
  - 1986, Ikujiro Nonaka en Hirotaka Takeuchi
- **Dynamic Systems Development Method (DSDM)**
  - 1995, DSDM Consortium
- **eXtreme Programming (XP)**
  - 1996, Kent Beck, Ken Auer, Ward Cunningham, a.o.
- **Feature Driven Development (FDD)**
  - 1997, Jeff de Luca

# About agile - Agile Manifesto (2001)

- Individuals and interactions *over* processes and tools
- Working software *over* comprehensive documentation
- Customer collaboration *over* contract negotiation
- Responding to change *over* following a plan

While there is value in the items on the right, the items on the left are valued *more!*

# Vision on testing in agile environments

- Manifesto is starting point
- Integrated test process
  - testing activities must be integrated in both the development process itself and in the team
  - testing should move the project forward
  - testing tools are increasingly important
  - testing is part of (definition of) done
- Find balance by making well-considered choices
- (Re)use values of proven test approaches

## Put the test vision into practice by:

- Seeing testing as the driver of the project
- Integrating testing in the entire process
- Start testing from the beginning of the project
- Integrating testing in the team
- Giving testing a role in the requirements elicitation
- Making it possible to test with a limited test basis
- Seeing testing as a part of "done"

# First things to do (being a tester)

- Make sure you understand the business (domain knowledge)
- Involve yourself in unit testing
- Assist other team members in non-testing activities
- Communicate about what is not documented
- Design test cases so they can be automated

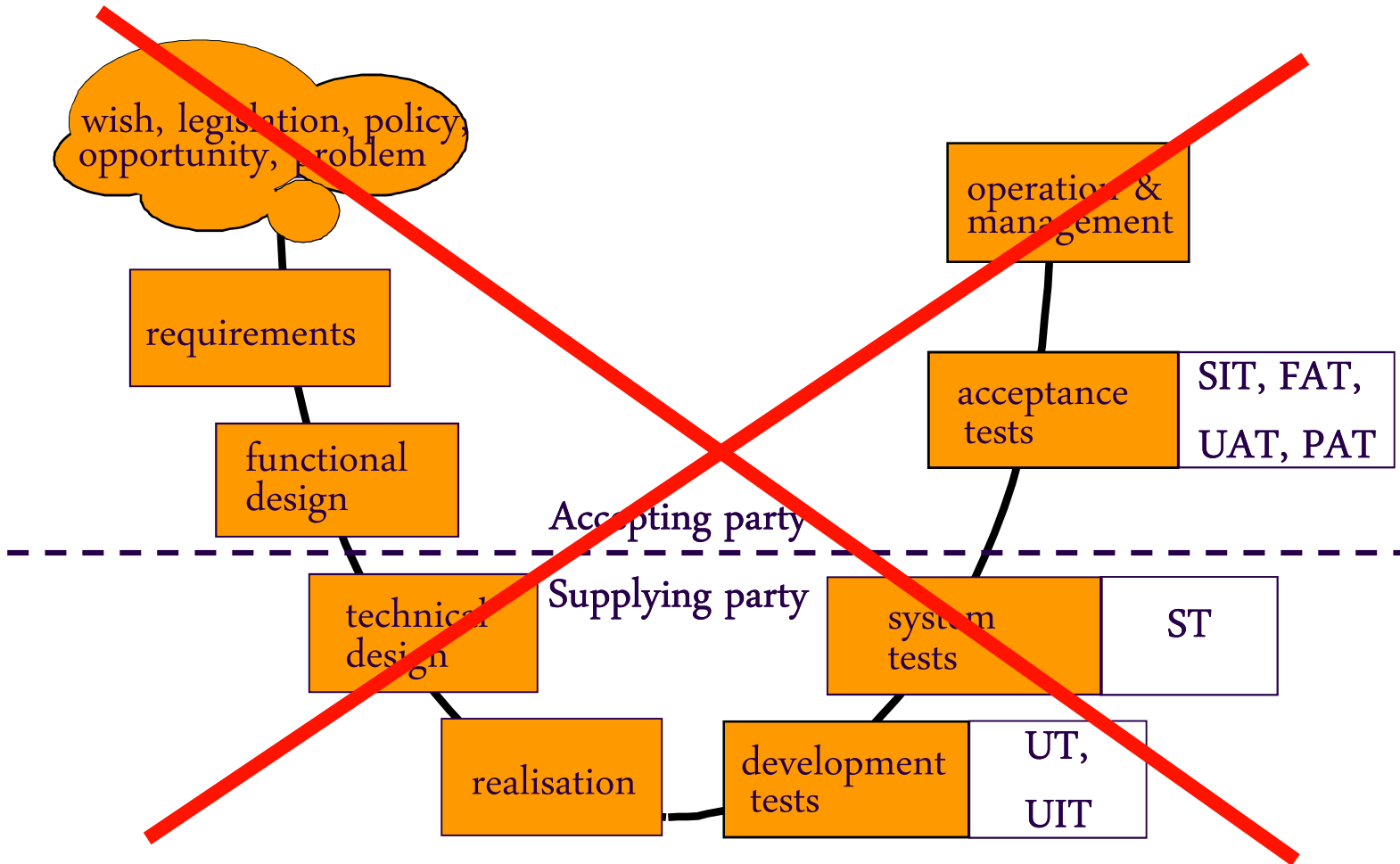


## No test levels needed - Characteristics

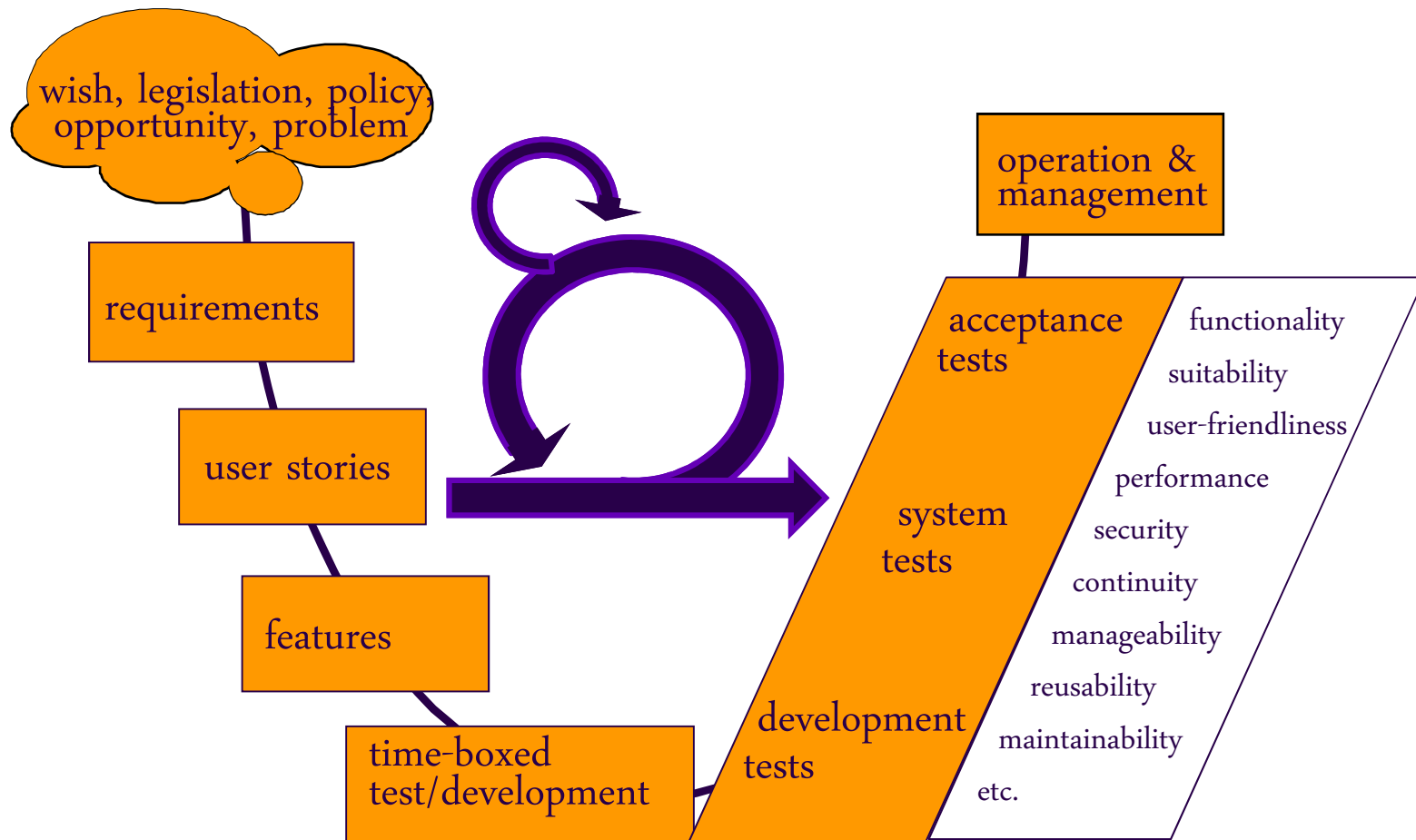
- No separate testlevels like system test or (user/production) acceptance test
- All team members work together
- All disciplines work together, support each other
- No designer, developer, user or test teams
- No test levels/teams with their own budget
- Team members accept the feature with their own acceptance criteria in mind



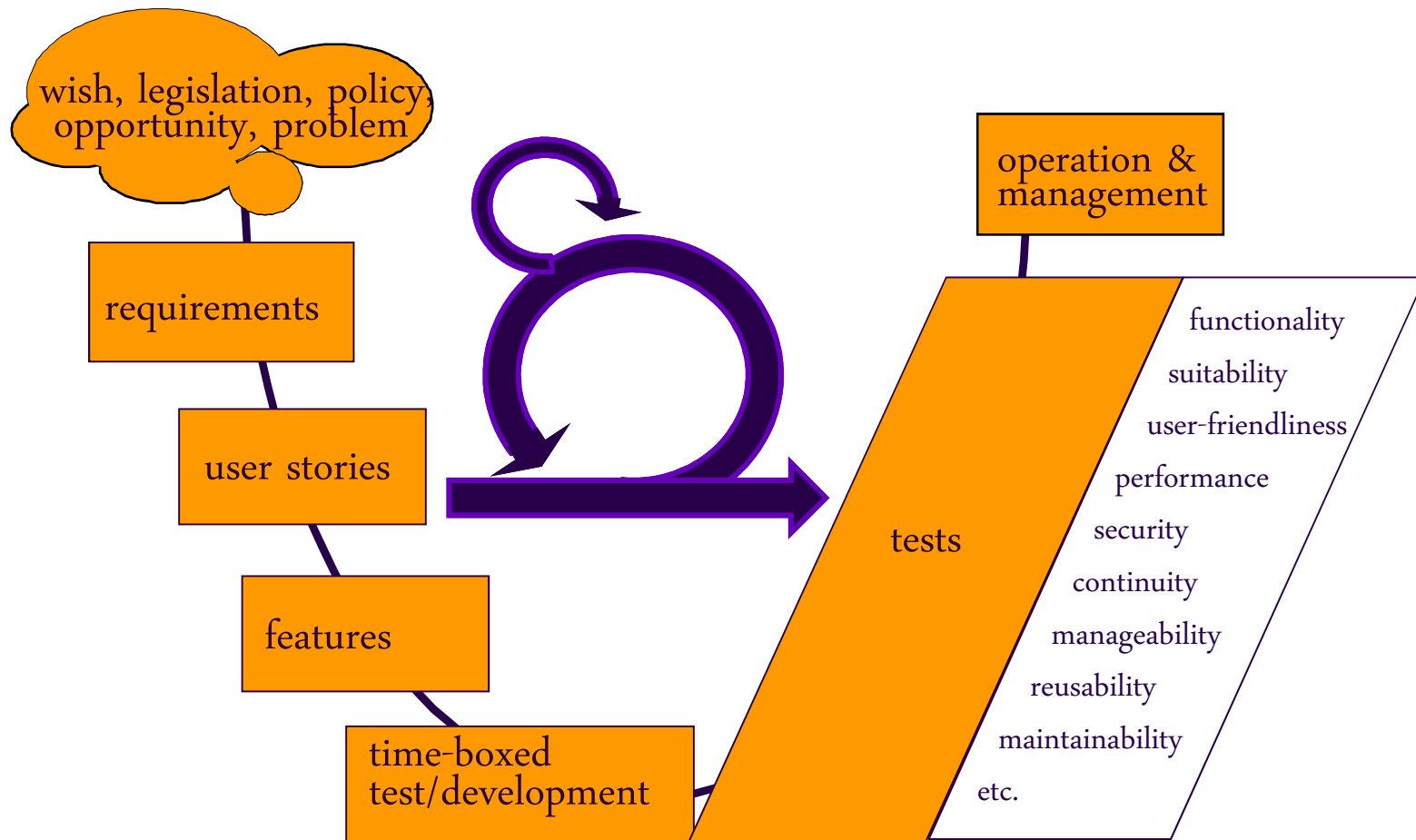
# No test levels needed - Traditionally



# No test levels needed – Intermediate



# No test levels needed – Agile way



# Product risk analysis – Stakeholders

User Story	Characteristic	Stakeholder
US 1	Functionality	Person A
	User-friendliness	Person B
US 2	Functionality	Person A
	Security	Person C
US 3	Functionality	Person A
US 4	Performance	Person C
US 5	Performance	Person C
US 6	Functionality	Person A
	Suitability	Person B
..	..	

# Product risk analysis – Risk table

User Story	Characteristic	Stakeholder	Damage	Chance of Failure	Risk Class
US 1	Functionality	Person A	3	3	9
	User-friendliness	Person B	2	1	2
US 2	Functionality	Person A	2	2	4
	Security	Person C	3	2	6
US 3	Functionality	Person A	2	1	2
US 4	Performance	Person C	2	1	2
US 5	Performance	Person C	1	1	1
US 6	Functionality	Person A	2	2	4
	Suitability	Person B	2	2	4
..	..		..	..	..

# Product risk analysis – Strategy table

User Story	Characteristic	Stakeholder	Damage	Chance of Failure	Risk Class	Intensity	Test Design Technique
US 1	Functionality	Person A	3	3	9	●●●	ECT-MCC
	User-friendliness	Person B	2	1	2	●	SYN
US 2	Functionality	Person A	2	2	4	●●	ECT-MCDC
	Security	Person C	3	2	6	●●	SEM-MCDC
US 3	Functionality	Person A	2	1	2	●	DCoT-EQ
US 4	Performance	Person C	2	1	2	●	EG
US 5	Performance	Person C	1	1	1	●	EG
US 6	Functionality	Person A	2	2	4	●●	ECT-MCDC
	Suitability	Person B	2	2	4	●●	PCT-TDL2
..	..		..	..	..		

# Product risk analysis – Progress table

User Story	Characteristic	S H	D	C o F	R C	Intensity	Test Design Technique	Test Cases created (Y/N)	Test Cases executed (Y/N)	Tests Passed (Y/N)
US 1	Functionality	A	3	3	9	●●●	ECT-MCC			
	User-friendliness	B	2	1	2	●	SYN			
US 2	Functionality	A	2	2	4	●●	ECT-MCDC			
	Security	C	3	2	6	●●	SEM-MCDC			
US 3	Functionality	A	2	1	2	●	DCoT-EQ			
US 4	Performance	C	2	1	2	●	EG			
US 5	Performance	C	1	1	1	●	EG			
US 6	Functionality	A	2	2	4	●●	ECT-MCDC			
	Suitability	B	2	2	4	●●	PCT-TDL2			
..	..		..	..	..					



- No separate test levels
- All acceptors present in relevant iterations
  - prepare/execute their own tests (with/without help)
- Risk determined per feature/qc by and per acceptor
- Test strategy determined per feature/qc
  - test intensity / test design techniques
- Just one table on white board
- Product is explicitly accepted by all acceptors



University of Applied Sciences

Thank you for your attention!

29TH ANNUAL  
PACIFIC NW  
SOFTWARE  
QUALITY  
CONFERENCE  
OCTOBER 10-12, 2011

Questions?



- **Email**

- [l.vanderaalst@fontys.nl](mailto:l.vanderaalst@fontys.nl)
- [info@leovanderaalst.nl](mailto:info@leovanderaalst.nl)

- **Phone**

- +31 (0)6 2277 8105

- **Internet**

- [www.fontys.nl/lectoren](http://www.fontys.nl/lectoren)
- [www.leovanderaalst.nl/home\\_eng.htm](http://www.leovanderaalst.nl/home_eng.htm)

