

# Agile Software Development

@ TNG Technology Consulting

2014 September 22  
AG Meeting 2014, Bamberg,  
Dr. Manfred Hanke,  
Senior Consultant

# Agile Software Development @ TNG

- TNG Technology Consulting
- Agile Software Development
- Live Demos: TDD & CI
- Conclusions
- Q & A

I ...

- ... studied physics in Regensburg and Erlangen
- ... graduated in astrophysics at ECAP / Remeis Observatory
  - PhD: “Probing the Environment of Accreting Compact Objects”
  - X-ray astronomy = data analysis and interpretation
    - requires quite some programming & automation:  
data reduction, modelling, computation of confidence intervals

I ...

- ... studied physics in Regensburg and Erlangen
- ... graduated in astrophysics at ECAP / Remeis Observatory
  - PhD: “Probing the Environment of Accreting Compact Objects”
  - X-ray astronomy = data analysis and interpretation
    - requires quite some programming & automation:  
data reduction, modelling, computation of confidence intervals
- ... once thought that I'd never want to become a consultant

I ...

- ... studied physics in Regensburg and Erlangen
- ... graduated in astrophysics at ECAP / Remeis Observatory
  - PhD: “Probing the Environment of Accreting Compact Objects”
  - X-ray astronomy = data analysis and interpretation
    - requires quite some programming & automation:  
data reduction, modelling, computation of confidence intervals
- ... once thought that I'd never want to become a consultant
- ... visited TNG on an ‘Open Techday’
  - and was really impressed by the spirit of that company!
- ... started in 2011 for TNG as a Software Consultant

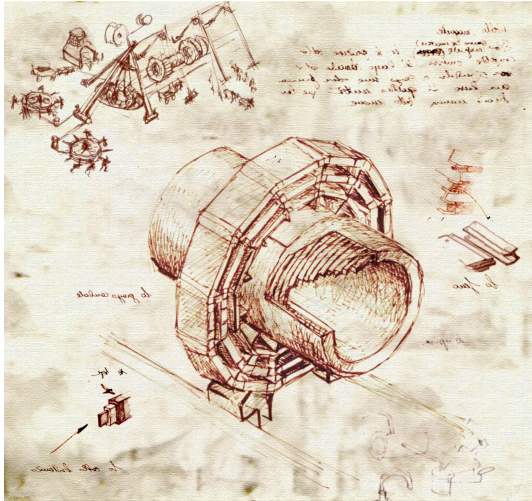
## TNG ...

- ... is a value-based consulting partnership, focused on high-end information technology
  - founded in 2001 aiming to create an optimal corporate structure
  - entirely self-financed (no bank loans; equity capital: 1 M€)
- ... has reached >150 employees (as of Aug 2014)
  - of which >98% hold a university degree and ~50% a PhD
  - annual growth rate: ~20%
- ... was amongst “Bayern’s Best 50” in 2010 and 2012
- ... provides know-how and support in three main areas:
  - (~70 %) Agile Software Development
  - (~15 %) Administration and Operations
  - (~15 %) IT Management

# How does one commonly build complex things?

# How does one commonly build complex things?

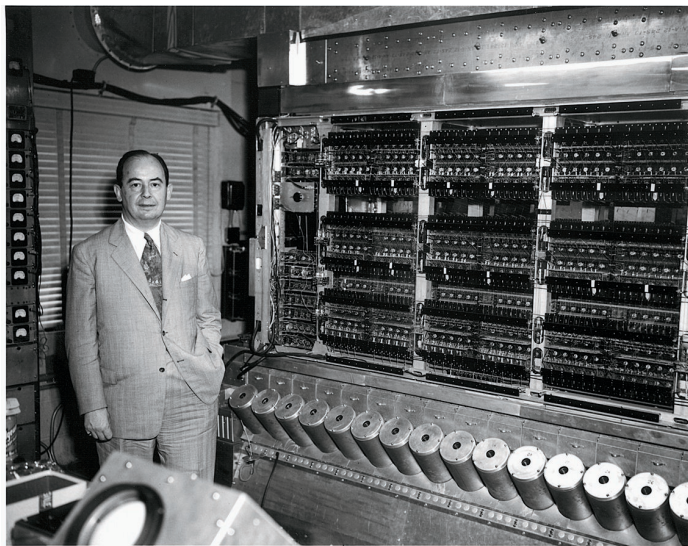
- One starts with a detailed construction plan.



CMS detector by Sergio Cittolin, © 2009 CERN



Unlike *hardware*, software is flexible!



John von Neumann & the IAS computer, 1952

## Software requirements are flexible, too...

- Desired features often cannot be fully specified in advance.
  - A client may just have a vision of a product.
- Business needs can change.
  - Software interacts with humans.
  - IT evolves quickly.

## Software requirements are flexible, too...

- Desired features often cannot be fully specified in advance.
  - A client may just have a vision of a product.
- Business needs can change.
  - Software interacts with humans.
  - IT evolves quickly.

” Some principles of software engineering:

- Humphrey's law (1995)  
*“For a new software system, the requirements will not be completely known until after the users have used it.”*
- Ziv's law (1996)  
*“Specifications will never be fully understood.”*
- Wegner's lemma (1997)  
*“An interactive system can never be fully specified nor can it ever be fully tested.”*

## Manifesto for Agile Software Development (2001)

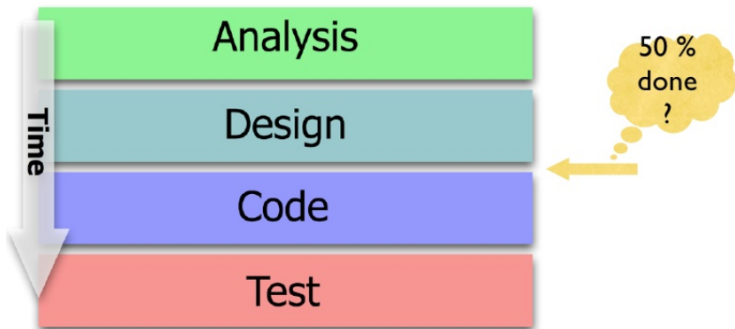
- *“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”*

## Manifesto for Agile Software Development (2001)

- *“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”*
- Values
  - Individuals and interactions  
over processes and tools
  - Working software  
over comprehensive documentation
  - Customer collaboration  
over contract negotiation
  - Responding to change  
over following a plan

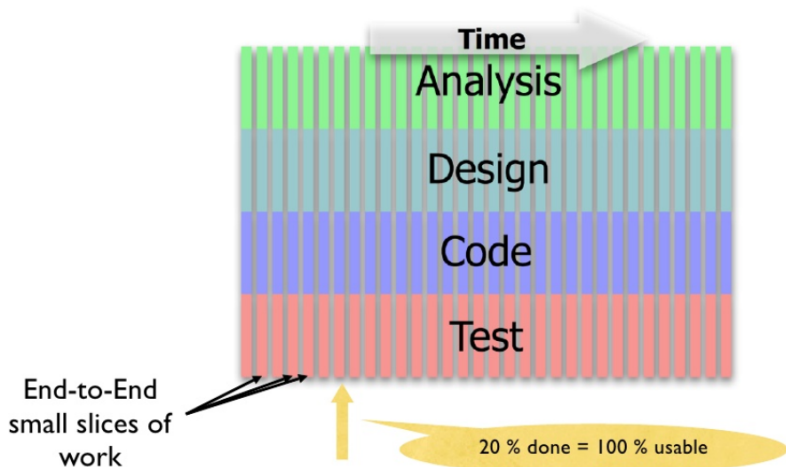
# Software development needs an iterative process

Traditional (“waterfall”) process:



from Agile Overview by Naresh Jain, licensed under Creative Commons

## Software development needs an iterative process

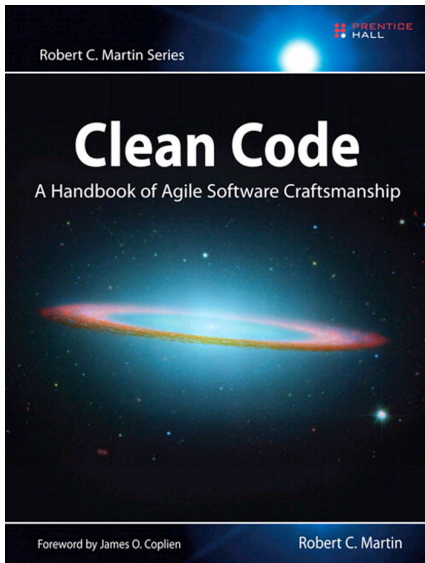


from Agile Overview by Naresh Jain, licensed under Creative Commons

The only way to be able to change code continuously



The only way to be able to change code continuously



## Automated software tests

- ... verify (repeatably) that code works as expected
  - when it is implemented
  - when it is refactored
  - when its dependencies are changed
- ... provide a live documentation of the code

## Automated software tests

- ... verify (repeatably) that code works as expected
  - when it is implemented
  - when it is refactored
  - when its dependencies are changed
- ... provide a live documentation of the code
- Example from StringUtils (Apache commons-lang library):

```
@Test public void testIsBlank() {  
    assertTrue(StringUtils.isBlank(null));  
    assertTrue(StringUtils.isBlank(""));  
    assertTrue(StringUtils.isBlank(" "));  
    assertFalse(StringUtils.isBlank("bob"));  
    assertFalse(StringUtils.isBlank(" bob "));  
}
```

## Test-driven development

- Cycle:
  1. Write a failing test for a new feature.
  2. Write just enough code to pass the test.
  3. Refactor the code to improve its design.
  
- Advantages
  - focused problem solving
  - simple code with convenient interfaces
  - valuable tests with a large code coverage

## Test-driven development

- Cycle:
  1. Write a failing test for a new feature.
  2. Write just enough code to pass the test.
  3. Refactor the code to improve its design.
- Advantages
  - focused problem solving
  - simple code with convenient interfaces
  - valuable tests with a large code coverage
- Live-Demo



## Continuous integration

- Automated build of all relevant components, executed for each change to the source code.
- Tests give fast feedback on regression or integration problems.
- Each successful build produces potentially-shippable artifacts.

## Continuous integration

- Automated build of all relevant components, executed for each change to the source code.
- Tests give fast feedback on regression or integration problems.
- Each successful build produces potentially-shippable artifacts.
  
- Live-Demo



Jenkins

# Agile software development can be very satisfying

- Your work produces meaningful, visible results.
- You are actually able to make changes.
- You find solutions which get to the heart of problems.
- Writing software is creative in itself, but also involves lots of communication.
- Becoming better is part of an iterative agile process.



# TNG provides

- ... an effective, lightweight company
- ... an agile spirit, plenty of enthusiasm for IT and nerd culture
- ... continuous education
  - 'Techdays', conferences, workshops, reading groups, ...

# TNG provides

- ... an effective, lightweight company
- ... an agile spirit, plenty of enthusiasm for IT and nerd culture
- ... continuous education
  - 'Techdays', conferences, workshops, reading groups, ...

★ Visit us during an 'Open Techday' and find out!

- on each month's second Friday
- Betastraße 13a, 85774 Unterföhring
- In the summer, there is usually a barbecue on our roof terrace...

<http://www.tngtech.com>

**TNG** TECHNOLOGY  
CONSULTING

FOR  
CLIENTS

FOR  
APPLICANTS

TNG  
ABOUT US

DE  
Q



We Solve  
Hard IT Problems.

Or drop me an email:  
[Manfred.Hanke@tngtech.com](mailto:Manfred.Hanke@tngtech.com)