A Quick Overview of Software Engineering

Paul Klint

nrc.nl - Economie - Honderdduizenden belastingaangiftes onbruikbaar - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.nrc.nl/economie/article952

Google

NRC HANDELSBLAD

NRC Handelsblad - voor nieuws, opinie en achtergrond

Nieuws
Voorpagina
Binnenland
Buitenland
Europa
Economie
Sport
Wetenschap
Kunst
Media
Beeld + Geluid
Nrc.next
Digitale Editie

Opinie & Debat
Opinie
Columns
Discussie
Commentaar
Ik@nrc.nl

Achtergrond
Nieuwsthema
Krantenarchief
Wereldblogs
Expertblogs

Cultuur
Boeken
Film
Cultuuragenda

Ontspanning
Fokke & Sukke
Denksport

Economie

Honderdduizenden belastingaangiftes onbruikbaar

Gepubliceerd: 27 februari 2008 07:44 | Gewijzigd: 27 februari 2008 13:52

Door een onzer redacteuren

Den Haag, 27 febr. Ruim 730.000 burgers moeten hun belastingaangifte over de inkomsten in 2007 opnieuw invullen door fouten in het computerprogramma van de Belastingdienst.

Dat schrijft staatssecretaris Jan Kees de Jager (Financiën, CDA) vandaag in een brief aan de Tweede Kamer. De Jager heeft meteen opdracht gegeven tot reorganisatie van het automatiseringscentrum van de Belastingdienst.

De fouten zijn opgetreden bij de verwerking van aangiften inkomstenbelasting over 2007 die tot en met 22 februari dit jaar met het internetprogramma van de Belastingdienst zijn ingediend. Bepaalde velden in de aangiften bleken onbruikbaar door fouten in de programmatuur, aldus De Jager. Hierdoor waren de aangiften onvolledig en konden zij niet worden verwerkt.

De 730.000 betrokken burgers

Staatssecretaris Jan Kees de Jager
(Foto Peter Hilz)

Done

Apologies for quoting some Dutch newspapers

nrc.nl - Binnenland - Twijfel aan veiligheid ov-chipkaart - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.nrc.nl/binnenland/article8

Google

NRC HANDELSBLAD

NRC Handelsblad - voor nieuws, opinie en achtergrond

Nieuws
Voorpagina
Binnenland
Buitenland
Europa
Economie
Sport
Wetenschap
Kunst
Media
Beeld + Geluid
Nrc.next
Digitale Editie

Opinie & Debat
Opinie
Columns
Discussie
Commentaar
Ik@nrc.nl

Achtergrond
Nieuwsthema
Krantenarchief
Wereldblogs
Expertblogs

Cultuur
Boeken
Film
Cultuuragenda

Ontspanning
Fokke & Sukke
Denksport

Service

Binnenland

Poortjes met chipkaart-scanners op station Amsterdam Zuid. *(Foto Peter Hilz)*

Twijfel aan veiligheid ov-chipkaart

Gepubliceerd: 9 januari 2008 14:16 | Gewijzigd: 9 januari 2008 14:22

Door een onzer redacteuren

Rotterdam, 9 Jan. TNO gaat in opdracht van Translink Systems de veiligheid van de ov-chipkaart onderzoeken. Dit omdat op een hackerscongres in Berlijn de zogeheten Mifare-technologie van de chipkaart gekraakt werd.

Het kraken gebeurde met apparatuur die voor een paar honderd euro te koop is.

Volgens Translink Sytems,

Nieuws - 'OV chipkaart toch begin 2009 invoeren'

Achtergrond - Toegangspoortje nekt invoering van ov-chipkaart

Done

Metro > Nieuws > Binnenland - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.metronieuws.nl/index.

Google

metro ..en fun!

INLOGGEN LEDEN | MELD JE AAN! | OVER CLUB METRO | KORTINGEN | PRIJSVRAGEN | AANBIEDING

NIEUWS

BINNENLAND
BUITENLAND
SPORT
WEER
VERKEERSINFORMATIE
ECONOMIE
FILM
INTERNET
ENTERTAINMENT
REIZEN

METRO

OVER METRO
DOWNLOAD DE KRANT
ADVERTEREN IN METRO
METRO LIFE PANEL
WERKEN BIJ METRO
RSS FEEDS
METRO MICROBLOG

Het laatste nieuws

at kan ik met RSS doen?

AMESBEURS GAAT VERHUIZEN NAAR KEULEN

metro

Strijd om Nederlandse

'Storingen A73-tunnels sneller verhelpen'

Geplaatst op woensdag 27 februari 2008 om 20:04

ROERMOND (ANP) - Rijkswaterstaat stelt alles in het werk om de aanhoudende reeks storingen in de beide A73-tunnels bij Roermond sneller te verhelpen. Ook zullen weggebruikers eerder worden geïnformeerd over afsluitingen, zodat ze tijdig een andere route kunnen kiezen.

Dat heeft de hoogste baas van Rijkswaterstaat, Bert Keijts, woensdag toegezegd aan wethouder Gerard IJff van Roermond. Minister Camiel Eurlings (Verkeer) had Keijts naar het stadhuis in Roermond gestuurd om iets te doen aan de voortdurende problemen rond de tunnels. Tijdens het bezoek van Keijts ging de Roertunnel een half uur lang dicht toen een te hoge vrachtwagen daar probeerde in te rijden.

Laatste

wo Kor Hel
wo 'Sto ver
wo Haa 520
wo Nec zaa
wo lede
wo CPE doc
wo Ver Wit-
wo Wa nen
wo Min Dinl
wo Exc bela
wo Der Roc
wo 'Aa sep
wo Nec amp
wo Kar ove
wo Gro slui

meer bi

Laatste

Done

Software Engineering is about ...

- Building large software systems
- Using state-of-the-art scientific knowledge
- Using solid engineering practices
- Dealing with evolving software
- Controlling complexity
- Controlling quality

Societal Relevance

- All parts of society essentially depend on software:
  - Government (taxes, social security, ...)
  - Healthcare
  - Services (banking and insurance companies)
  - Transportation, logistics, telecommunication
  - Entertainment (games, animation)

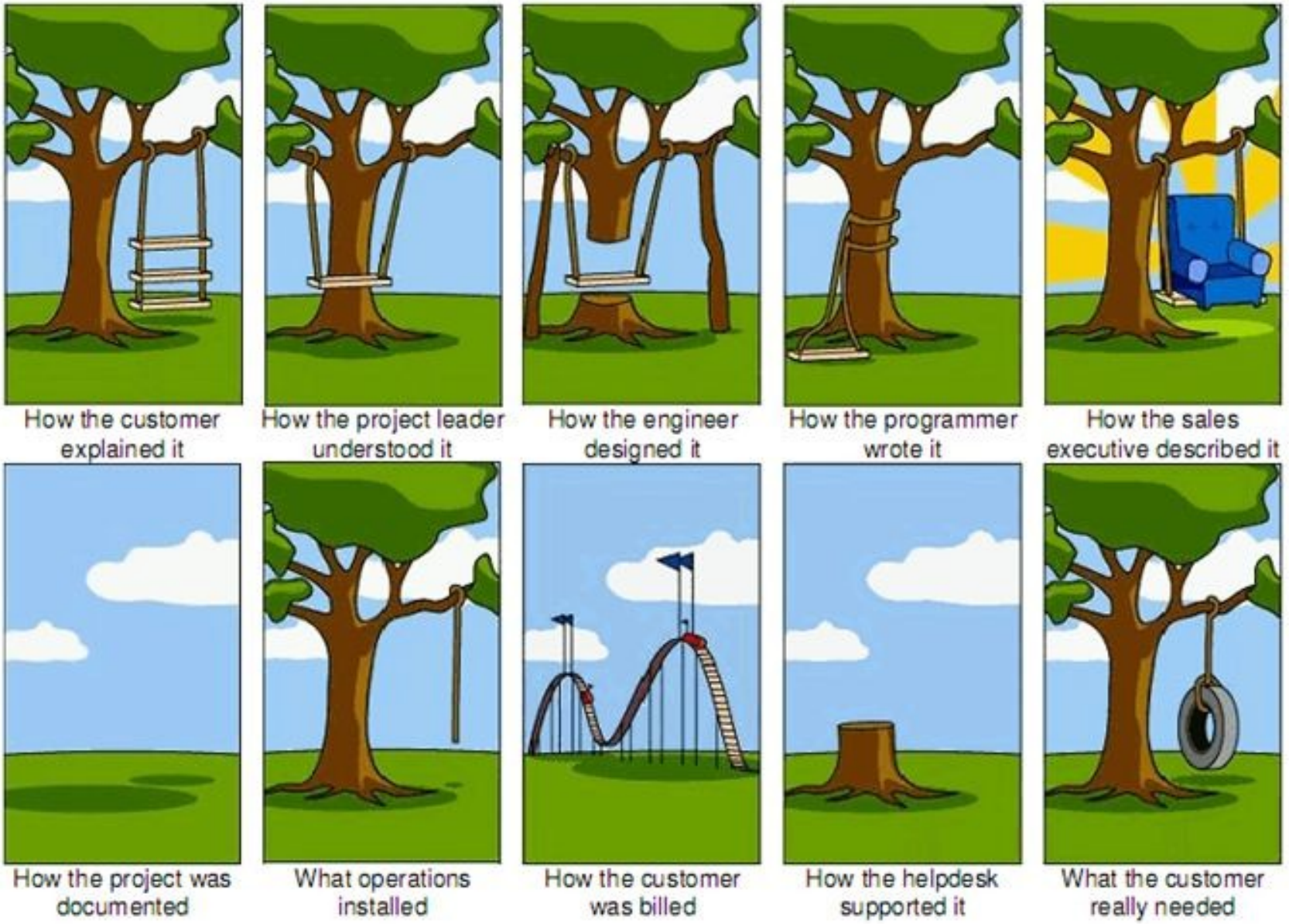
Facts about Software

- 50% of all software projects costs more than twice the original budget
- One-third of all projects has a delay of 200-300%
- 80% of all projects fails
- Failing software costs in the US 56 billion $ per year
- 60% of all programmers works on maintenance

Software Engineering has to reconcile

- The ***requirements*** of the stakeholders and users of a system
- The ***engineering*** of the system including designing, building, testing, and maintaining it
- The ***process*** how to organize all the above

Requirements



Requirements

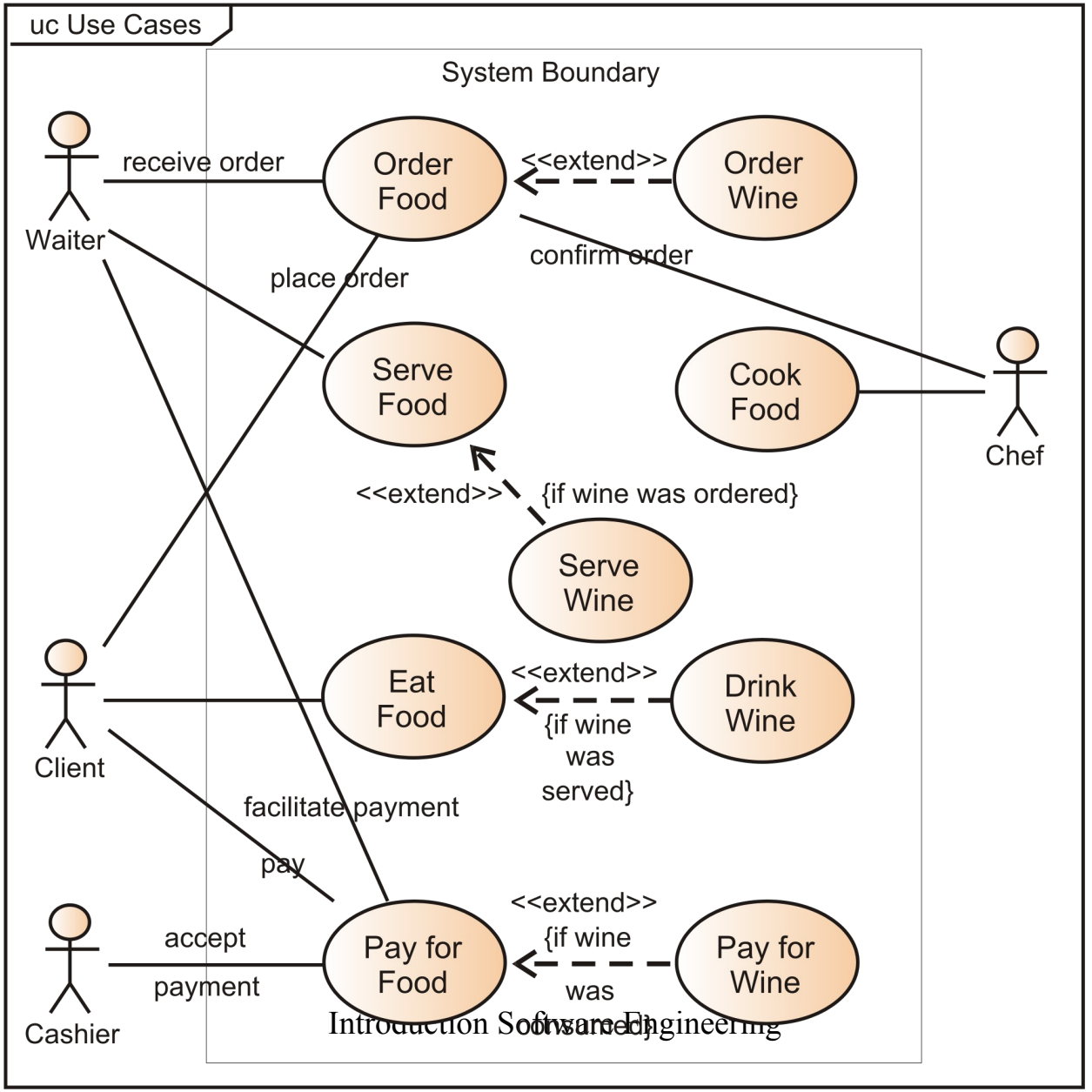
Requirements

- Interviews with stakeholders
- User stories and Use cases
- Mock ups
- Early prototypes
- “Customer on site”

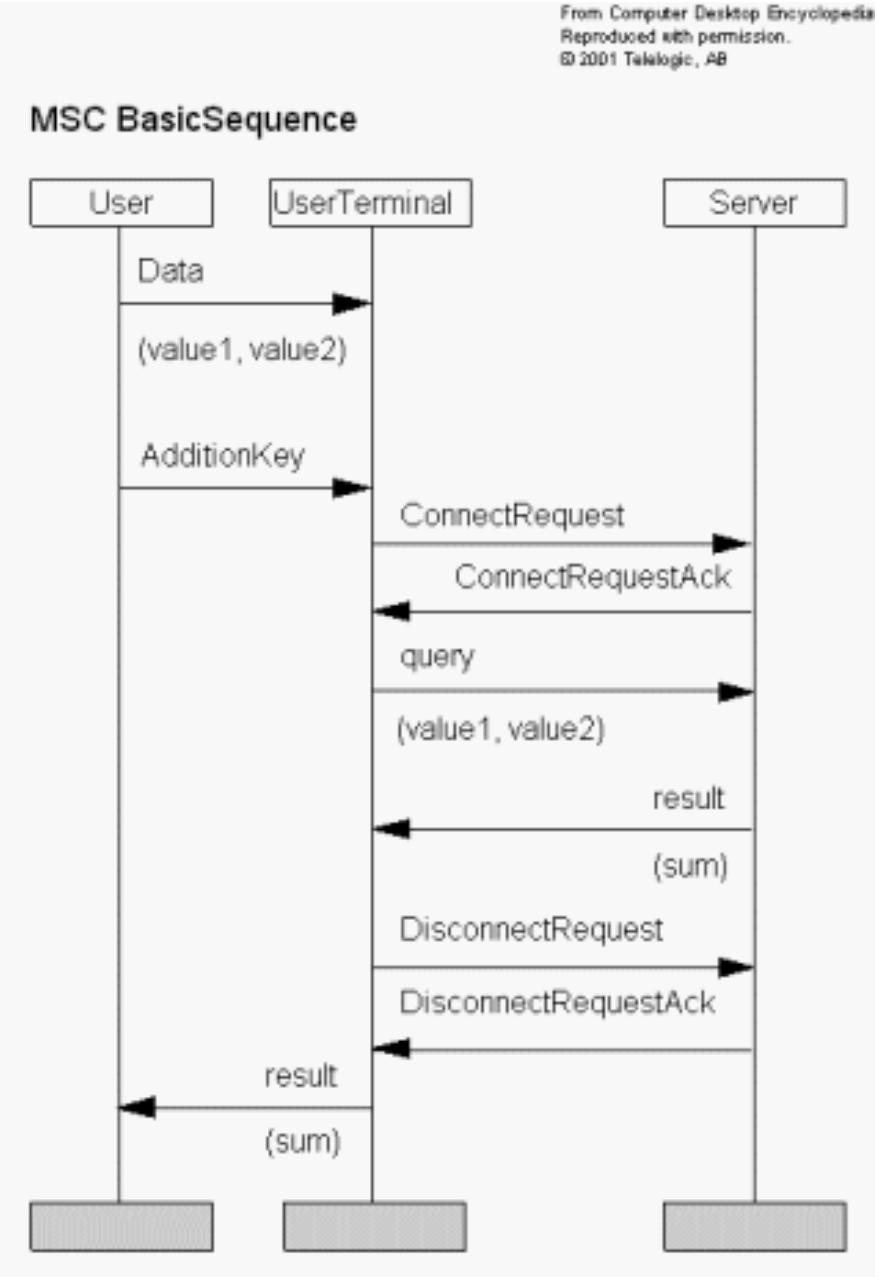
User Stories

- Pattern: `As a <role>,I want <goal/desire>`
- `As a user, I want to search for my customers by their first and last names.`
- `As a user closing the application, I want to be prompted to save if I have made any change in my data since the last save.`

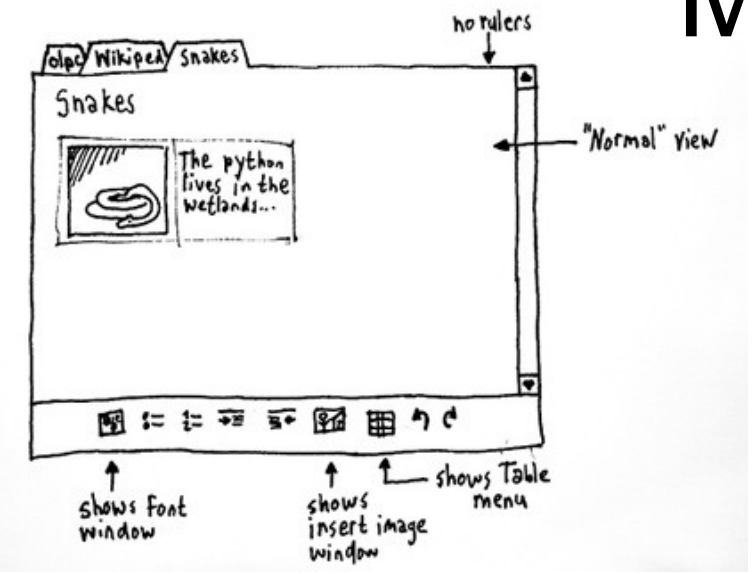
Use Case

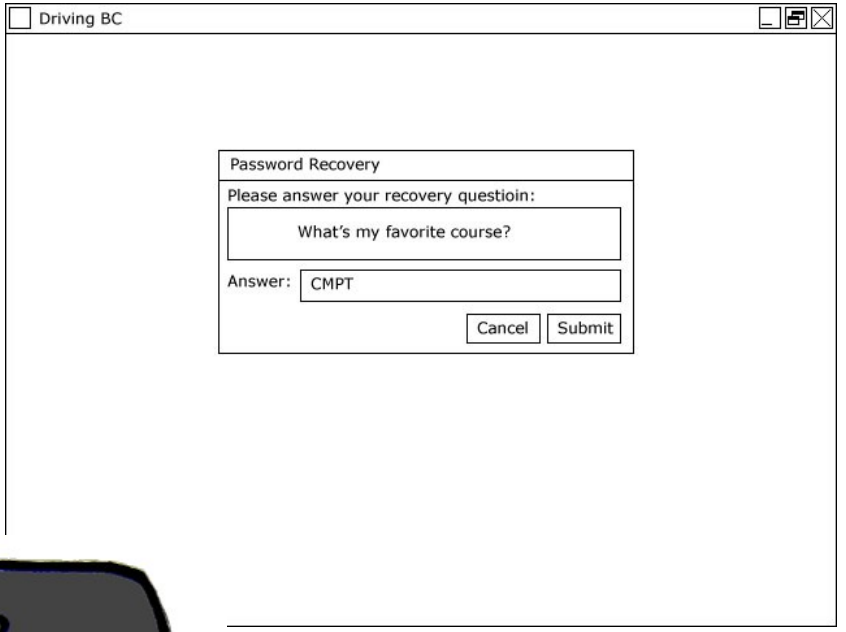

uc Use Cases
System Boundary
Waiter
receive order
Order Food
<<extend>>
Order Wine
confirm order
place order
Serve Food
Cook Food
Chef
<<extend>>
{if wine was ordered}
Serve Wine
Client
Eat Food
<<extend>>
{if wine was served}
Drink Wine
facilitate payment
pay
Cashier
accept payment
Pay for Food
<<extend>>
{if wine was ordered}
Pay for Wine

Message Sequence Chart



Mock-ups


no rulers
olpc
Wikiped
Snakes
Snakes
The python lives in the wetlands...
"Normal" view
shows font window
shows insert image window
shows Table menu


Driving BC
Password Recovery
Please answer your recovery questioin:
What's my favorite course?
Answer:
CMPT
Cancel
Submit


ABC 3G 03:42 PM
Settings
Wi-Fi Networks
Wi-Fi ON
Choose a Network...
Alice-09413358
Balsamiq
Untitled
Other
Ask to Join Network ON
Known networks will be joined automatically. If no known networks are available, you will be asked before joining a new network.
ABC 3G 03:42 PM
Settings
Start Screen Last Used
Search
Searchable Items
SafeSearch Moderate
Voice Search ON
Voice Search only works in English, and works best fro North American English accents.
Apps
Search
Apps
Settings
ABC 3G 03:51 PM
Friends
Scobleizer...
balsamiq...
timoreilly: RT ...
timoreilly...
Friends
Replies
Messages
Favorites
Search

Engineering

- Software Architecture
- Software Design
- Software Construction
- Software Quality Assessment
- Software Maintenance

Software Architecture and Design

Sagrada Familia
Barcelona, Spain,
1883 -- ...

Antoni Gaudi,
1852 -- 1926

Why is Gaudi's Design Good?

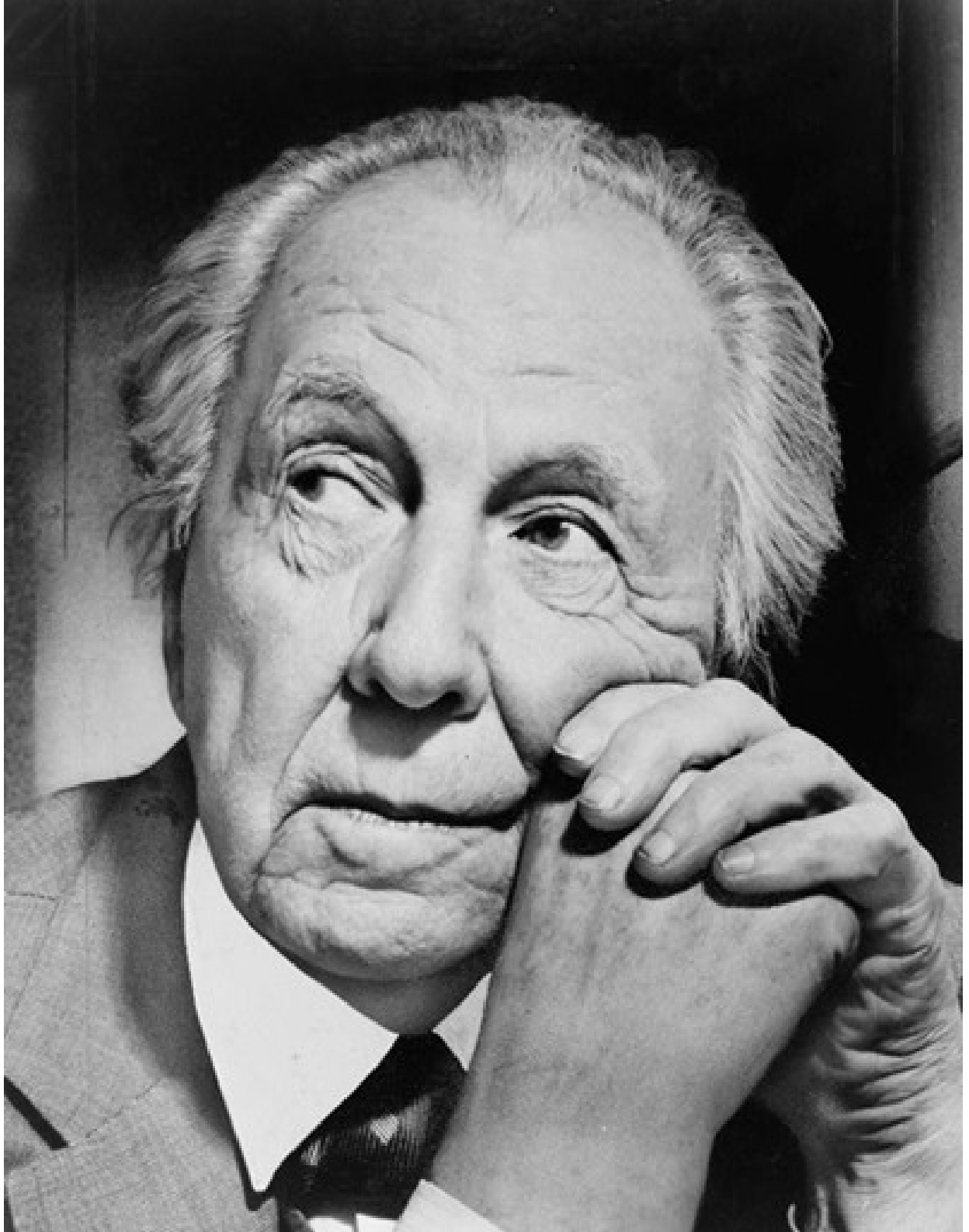
- Forms inspired by nature
- Organic integration of parts
- Each facade a separate theme
- Unconventional
- Impressive

Note: miniature models were used to study the design

Johnson Wax Headquarters 1936-1939


Fallingwater, 1935

Robie House 1908 --1910



Frank Loyd Wright
1867 -- 1959

Why are Loyd Wright's Designs Good?

- Clear lines and forms
- Simple
- Design has a perfect fit with the environment
- Careful consideration how light enters building

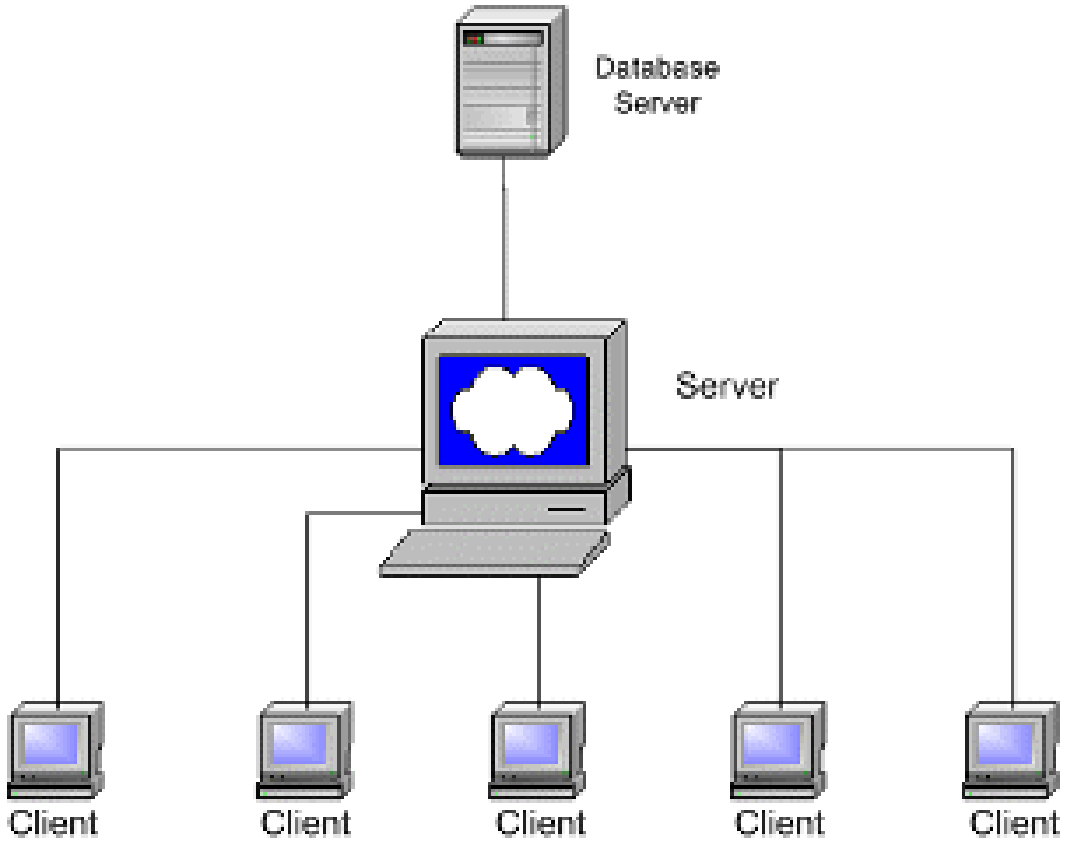
Gasuniegebouw, Groningen, Alberts en Huut,
“Mooiste gebouw van Nederland”, Trouw, 2007.

What is Software Architecture?

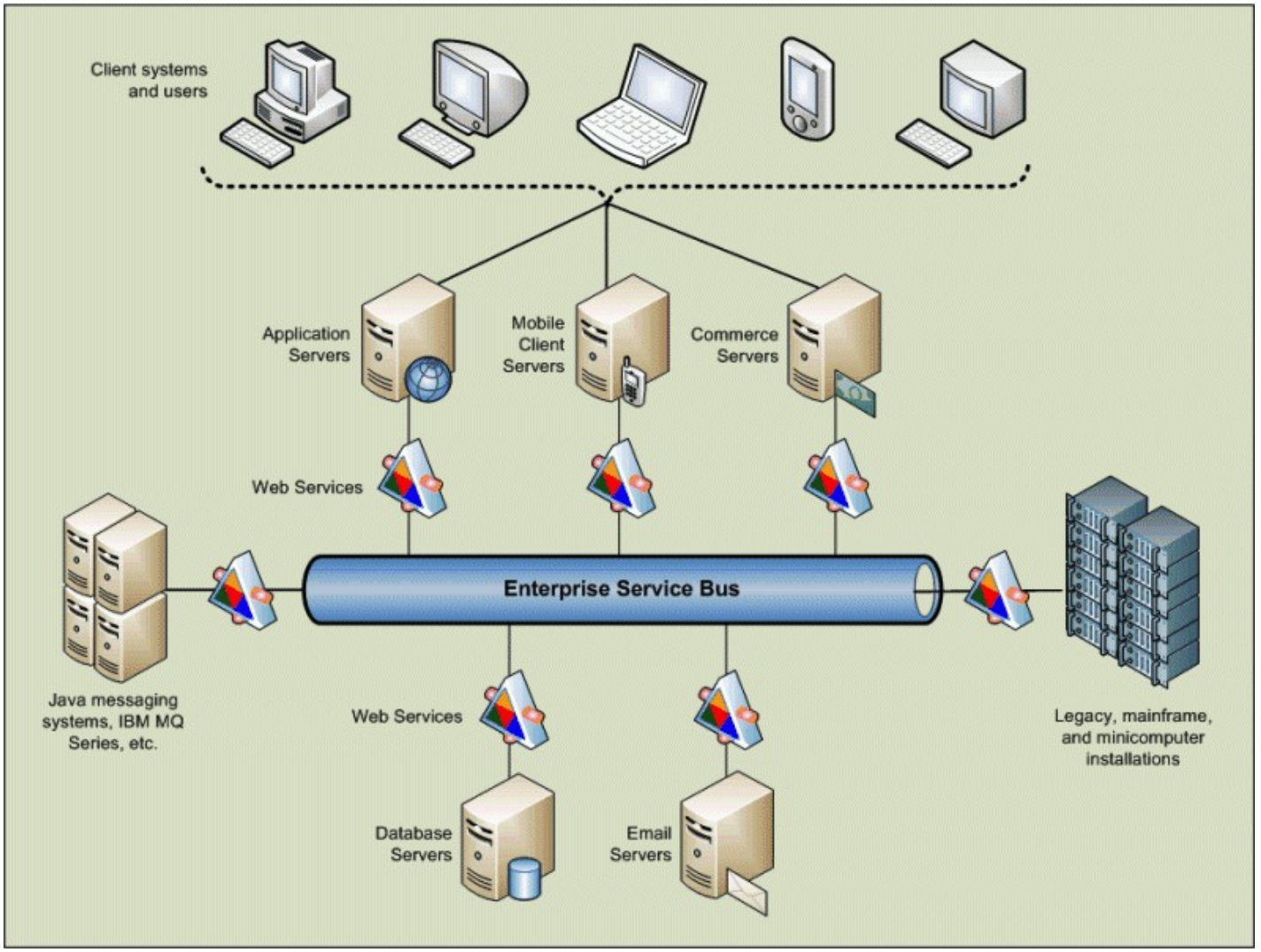
The **software architecture** of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both.

The term also refers to documentation of a system's "software architecture." Documenting software architecture facilitates communication between stakeholders, documents early decisions about high-level design, and allows reuse of design components and patterns between projects

Software Architecture
Client-server architecture



Software Architecture Service Bus



What is Software Design?

Software design is a process of problem-solving and planning for a software solution. After the purpose and specifications of software are determined, software developers will design or employ designers to develop a plan for a solution. It includes low-level component and algorithm implementation issues as well as the architectural view. *Source*: Wikipedia

Software Design

- Design Patterns
- Algorithm Design
- Data design
- Unclear separation between high-level and low-level concerns

SW Design is a “Wicked” Problem

- No definite formulation of the problem
- No stopping rule: we never reach “the” solution
- Solution not true/false but only good/bad. Many acceptable solutions
- Every solution may lead to other wicked problems
- *Source*: Van Vliet, Software Engineering

Global SW Design Strategies

Top-down: start from problem domain

- Pro: good fit with the requirements, global picture
- Con: unclear whether you can reach good (efficient) solution at the code level

Bottom-up: start from solutions domain

- Pro: good solution for local problems
- Con: unclear whether you will eventually satisfy the global requirements

YoYo: reconcile these strategies

YoYo Design

- Start with global, top-down, design
- Do bottom-up design of selected parts to verify that detailed design is feasible
- Integrate with (and change) global design
- Repeat

Software Construction

- Programming Languages
- Paradigms: imperative – OO – functional – ...
- Approaches:
  - Design Patterns
  - Domain-specific Languages (DSLs)
  - Model-driven
  - Aspect-oriented programming
- Tools: IDEs, debuggers, test tools, version management, ...

Software Quality Assessment

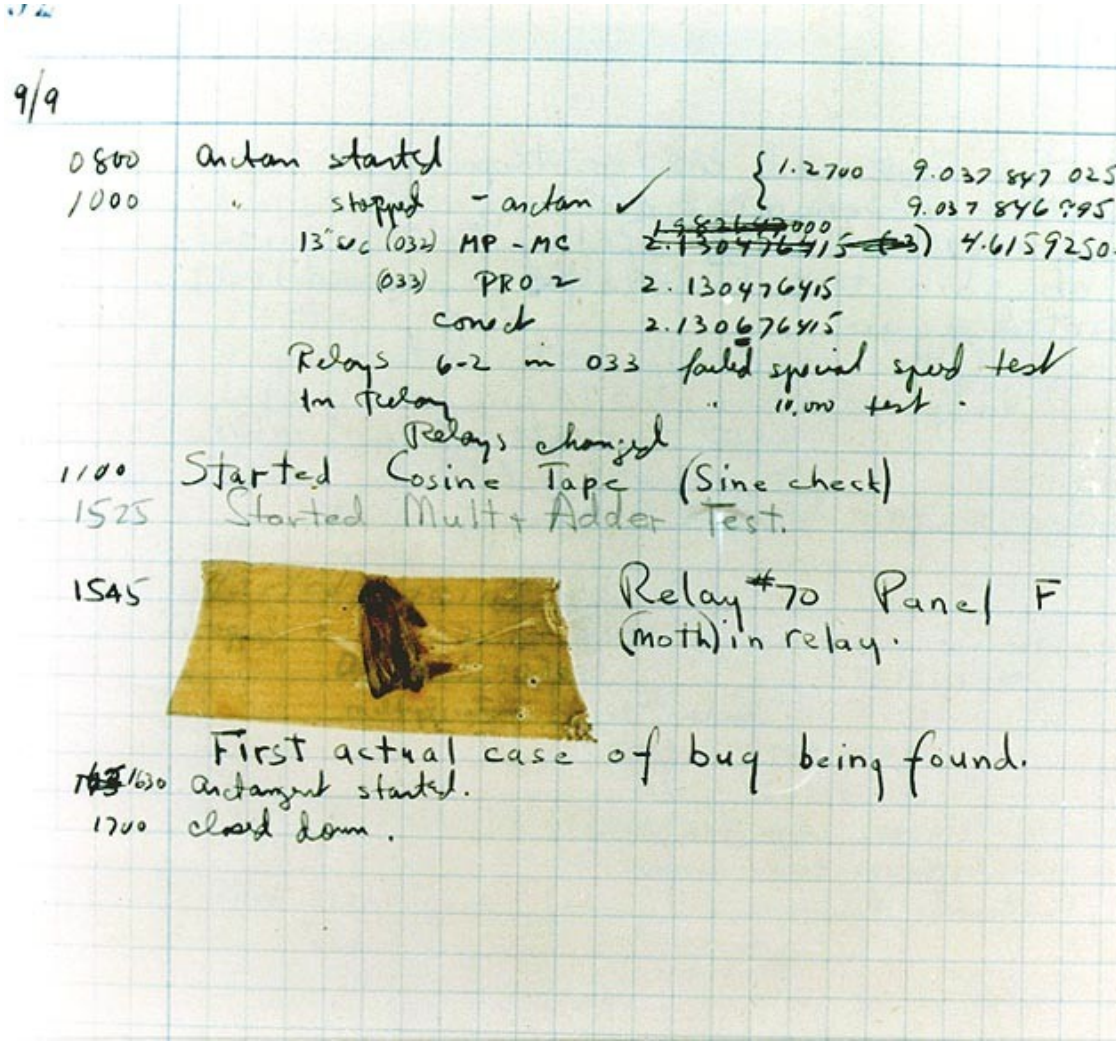
- Software complies with requirements and is functionally correct?
- Non-functional quality aspects:
  - Performance, reliability, security, maintainability, documentation, ease-of-use, ...
- Rule of thumb: the earlier a problem is found, the cheaper you can fix it:
  - Requirements (1x); Post-release (100x)

Code Review

- Interactive session, where a developer explains his/her code to colleagues
- Finds more errors than any other technique!
- Pair-programming (discussed later) supports this

The Word “bug”

- In relay-based machines little insects could disrupt machine operation
- Here a moth found in the Harvard Mark II
- Now used for software errors
- Debugging: removing errors



Testing

- "Testing can only show the presence of bugs, not their absence" (Edsger Dijkstra)
- White-box testing
  - During testing the implementation is known
- Black-box testing
  - Implementation is unknown
- Test levels
  - Unit
  - Integration
  - System

Manual Unit Testing

- Manually write test cases that exercise a function for average values and boundary values
- Example: test cases for a list sort function:
  - Empty list
  - Non-empty, sorted, lists
  - Non-empty, reversely sorted, lists
  - Random lists of various sizes

Automatic Unit Testing

- Write a predicate `isSorted` to test that a list is sorted.
- `bool testSort(list[int] L) = isSorted(sort(L))`
- Randomly generate lists and use `testSort` as oracle.
- This approach is called QuickCheck and is highly effective.

Test Tools for the Rascal programmer

- `test` is a function modifier, currently for Boolean functions without parameters.
- Use `:test` on the command line to run all test functions
- A Master's project is well-advanced to add QuickCheck functionality to Rascal's test framework.

Testing Rascal itself

- Thousands of tests for basic language elements and libraries
- Code examples in Tutor are also executed and also act as tests.
- Developer and tester should wear different “hats”

Testing the Integer Operators

```
@Test
public void testInt()
{
assertTrue(runTest("1 == 1;"));
assertTrue(runTest("1 != 2;"));

assertTrue(runTest("-1 == -1;"));
assertTrue(runTest("-1 != 1;"));

assertTrue(runTest("1 + 1 == 2;"));
assertTrue(runTest("-1 + 2 == 1;"));
assertTrue(runTest("1 + (-2) == -1;"));

assertTrue(runTest("2 - 1 == 1;"));
assertTrue(runTest("2 - 3 == -1;"));
assertTrue(runTest("2 - -1 == 3;"));
assertTrue(runTest("-2 - 1 == -3;"));

assertTrue(runTest("2 * 3 == 6;"));
assertTrue(runTest("-2 * 3 == -6;"));
assertTrue(runTest("2 * (-3) == -6;"));
assertTrue(runTest("-2 * (-3) == 6;"));

assertTrue(runTest("8 / 4 == 2;"));
assertTrue(runTest("-8 / 4 == -2;"));
assertTrue(runTest("8 / -4 == -2;"));
assertTrue(runTest("-8 / -4 == 2;"));

assertTrue(runTest("7 / 2 == 3;"));
assertTrue(runTest("-7 / 2 == -3;"));
assertTrue(runTest("7 / -2 == -3;"));
assertTrue(runTest("-7 / -2 == 3;"));

assertTrue(runTest("0 / 5 == 0;"));
assertTrue(runTest("5 / 1 == 5;"));

assertTrue(runTest("5 % 2 == 1;"));
assertTrue(runTest("-5 % 2 == -1;"));
assertTrue(runTest("5 % -2 == 1;"));

assertTrue(runTest("-2 <= -1;"));
assertTrue(runTest("-2 <= 1;"));
assertTrue(runTest("1 <= 2;"));
assertTrue(runTest("2 <= 2;"));
assertFalse(runTest("2 <= 1;"));

assertTrue(runTest("-2 < -1;"));
assertTrue(runTest("-2 < 1;"));
assertTrue(runTest("1 < 2;"));
assertFalse(runTest("2 < 2;"));

assertTrue(runTest("-1 >= -2;"));
assertTrue(runTest("1 >= -1;"));
assertTrue(runTest("2 >= 1;"));
assertTrue(runTest("2 >= 2;"));
assertFalse(runTest("1 >= 2;"));

assertTrue(runTest("-1 > -2;"));
assertTrue(runTest("1 > -1;"));
assertTrue(runTest("2 > 1;"));
assertFalse(runTest("2 > 2;"));
assertFalse(runTest("1 > 2;"));

assertTrue(runTest("(3 > 2 ? 3 : 2) == 3;"));

}
```

Testing Sort

```
@Test
public void sort() {

prepare("import List;");

assertTrue(runTestInSameEvaluator("{List::sort([]) == [];}"));
assertTrue(runTestInSameEvaluator("{sort([]) == [];}"));
assertTrue(runTestInSameEvaluator("{List::sort([1]) == [1];}"));
assertTrue(runTestInSameEvaluator("{sort([1]) == [1];}"));
assertTrue(runTestInSameEvaluator("{List::sort([2, 1]) == [1,2];}"));
assertTrue(runTestInSameEvaluator("{sort([2, 1]) == [1,2];}"));
assertTrue(runTestInSameEvaluator("{List::sort([2,-1,4,-2,3]) == [-2,-1,2,3, 4];}"));
assertTrue(runTestInSameEvaluator("{sort([2,-1,4,-2,3]) == [-2,-1,2,3, 4];}"));
}
```

Test Driven Developmen (TDD)

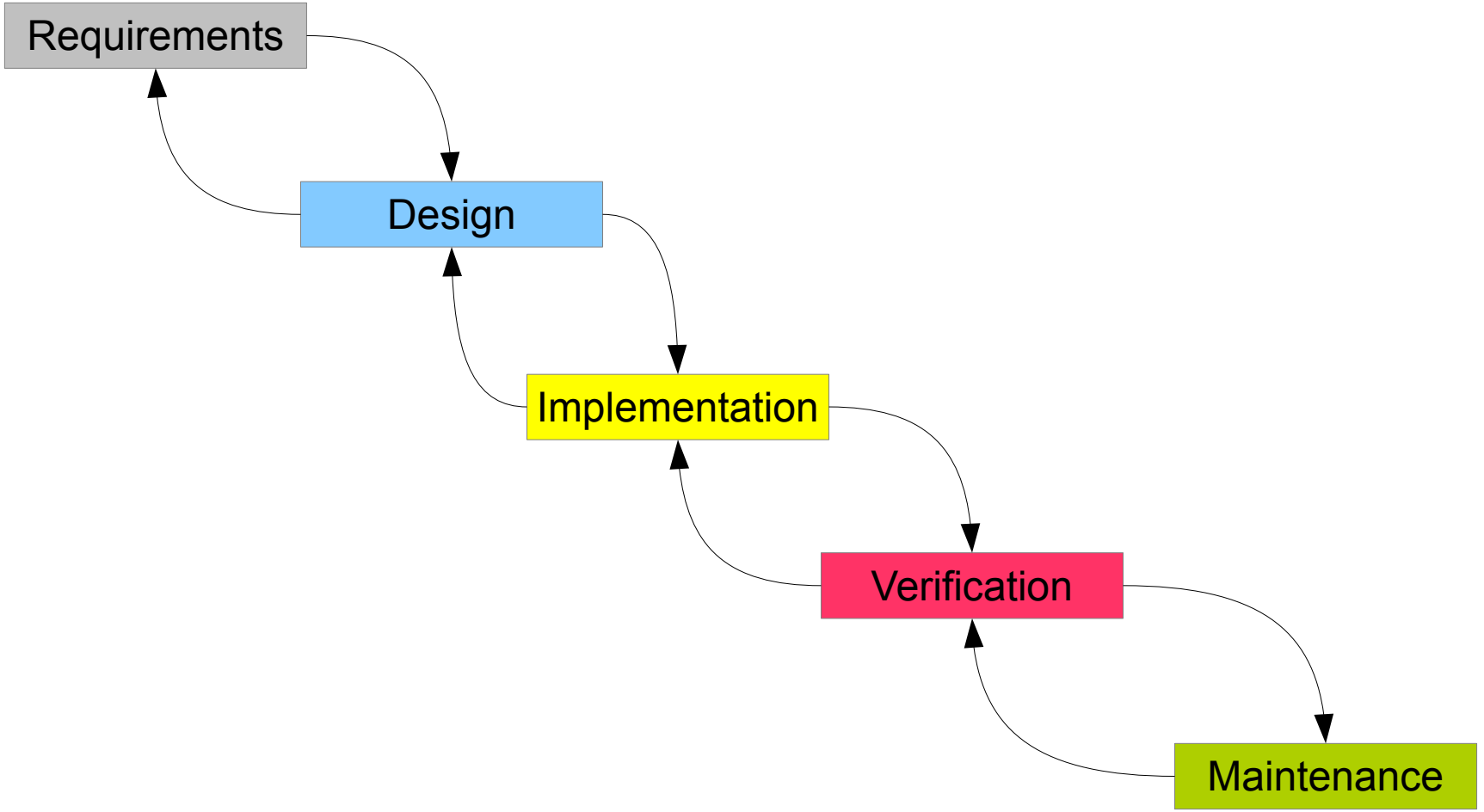
- Write test cases first (they act as a specification)
- Then write the implementation of the code that is tested

The Software Process

Software Process

- All aspects of the software ***life cycle***
- Two main styles:
  - Waterfall
  - Agile

Waterfall Model



Waterfall Model

- All steps are traceable
- Solid engineering practices
- Long development cycles
- Not so easy to steer the project
- Much bureaucracy

Agile Development

- Originally called Extreme Programming
- Agile Manifesto
- Agile Principles
- Agile Practices
- Large commercial interests in “selling” you agile methods

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

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

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Agile Principles

- Customer satisfaction by rapid delivery of useful software
- Welcome changing requirements, even late in development
- Working software is delivered frequently (weeks, not months)
- Working software is the principal measure of progress
- Sustainable development, able to maintain a constant pace
- Daily co-operation between business people and developers
- Face-to-face conversation is the best form of communication (co-location)
- Projects are built around motivated, trusted, individuals
- Continuous attention to technical excellence and good design
- Simplicity- The art of maximizing the amount of work not done - is essential
- Self-organizing teams
- Regular adaptation to changing circumstances

Agile Practices

- Pair programming
- Design patterns
- Continuous Integration
- Test-driven development
- Automated unit testing
- Refactoring

Agile Development

- Many agile approaches: SCRUM, Kanban, XP, DSDM
- There is no scientific proof that waterfall methods or agile methods are “better”
  - We have done various studies; it is hard to obtain and compare data
- The influence of individual programmers is large (10x difference between individuals)

Software Engineering

- At the core of
  - Web technologies
  - Mobile applications
  - Gaming
  - Big Data
  - Scientific computing
- Many faces:
  - Technical
  - Human
  - Organizational

CareerCast, a site for job seekers, listed the top 200 jobs of 2012. Software engineer had the best overall score when you average work environment, income, stress level, physical demands and hiring outlook. The site, which used Department of Labor statistics, pegged the average income of a software engineer at $88,142.

15 Rad Robot Accessories for Your Office 5

325 855

'Software Engineer' Tops List of 2012's Best Jobs

April 13, 2012 by **Todd Wasserman**

10

Do you code for a living? Then congratulations, you have the best job in America right now, according to a new survey.

CareerCast, a site for job seekers, listed the top 200 jobs of 2012. Software engineer had the best overall score when you average work environment, income, stress level, physical demands and hiring outlook. The site, which used Department of Labor statistics, pegged the average income of a software engineer at $88,142.