Technical University of Munich| Faculty of Informatics

Chair of Computer Aided Medical Procedures and Augmented Reality

Lab Course / "Praktikum": *Project Management and Software Development for Medical Applications*

Documentation, Tests, Design Patterns & Integration Strategy – SS2022

Conducted by:
Ardit Ramadani, Lennart Bastian and **Tianyu Song**
Prof. Dr. Nassir Navab - CAMP

Munich, 10 May 2022

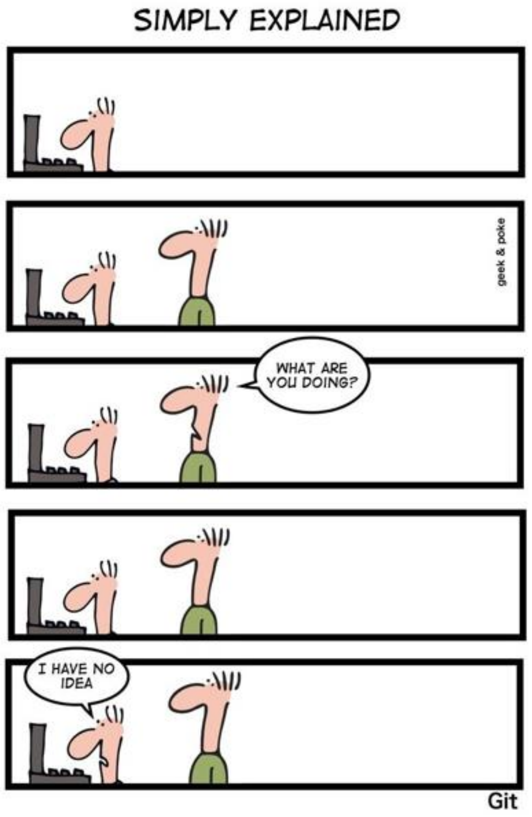
Technische Universität München

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Slide courtesy of Marco Esposito, Rüdiger Göbl and Dr. Benjamin Frisch

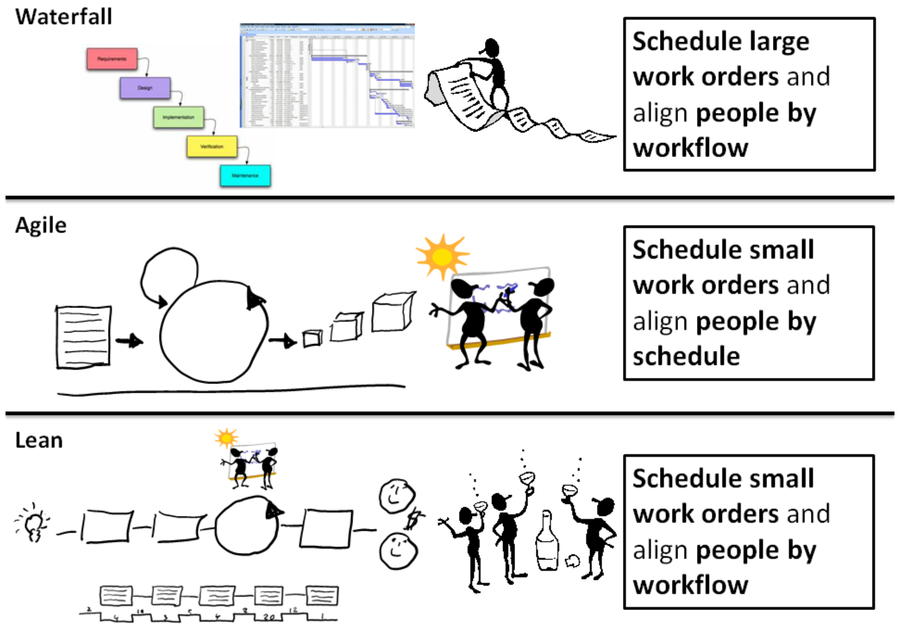
Disclaimer

- This talk will not cover all aspects of SE!
- Familiarize with concepts and ideas
- Not every single detail matters



Software Engineering approaches

- Sometimes it is applied rigidly
- Many different contrasting ideas
- Do not get your attention drawn away from the problem at hand!

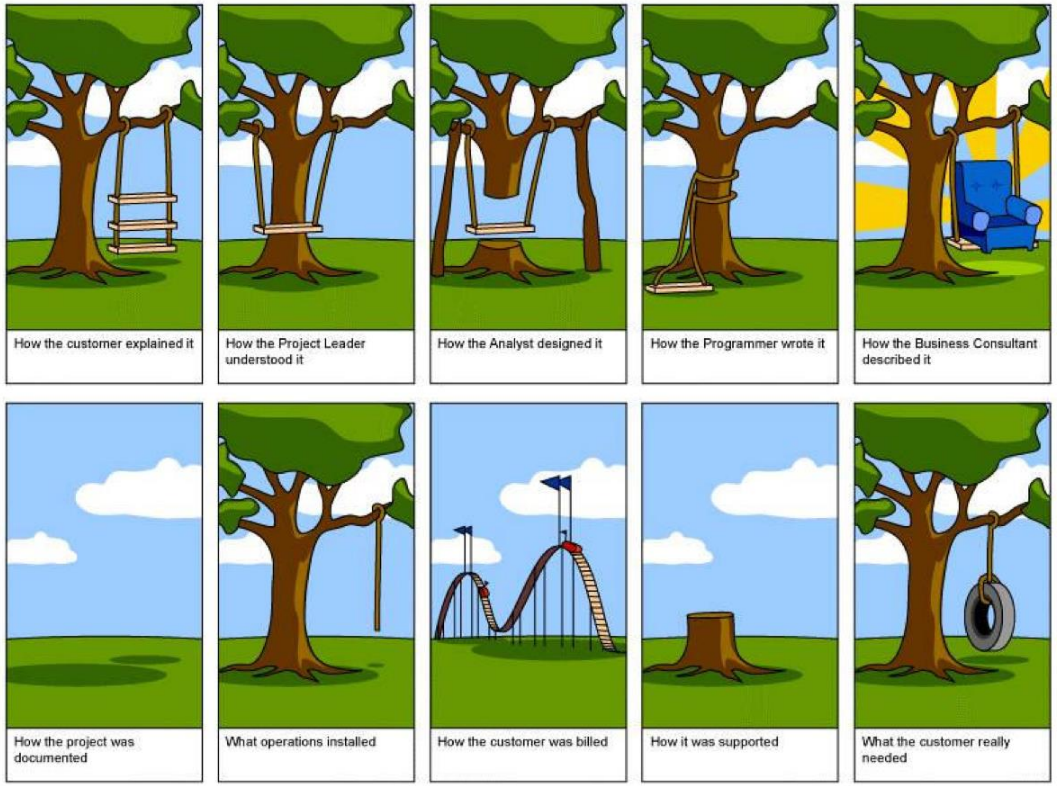


Schedule large work orders and align people by workflow

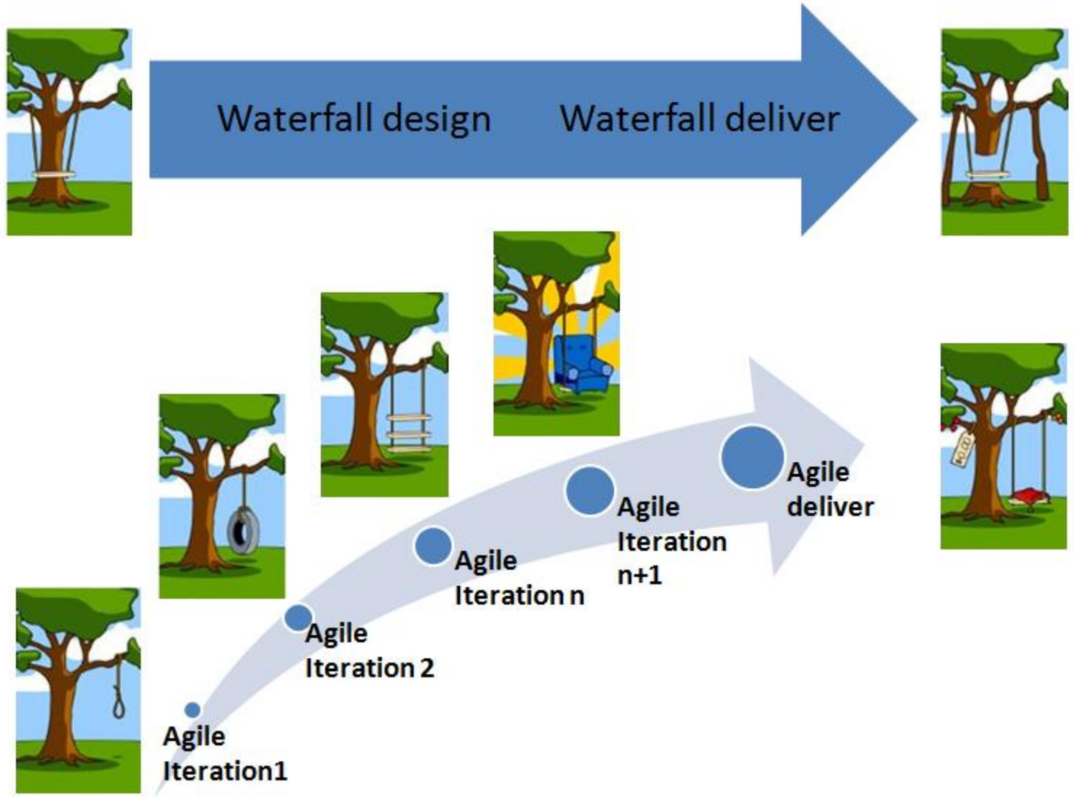
Schedule small work orders and align people by schedule

Schedule small work orders and align people by workflow

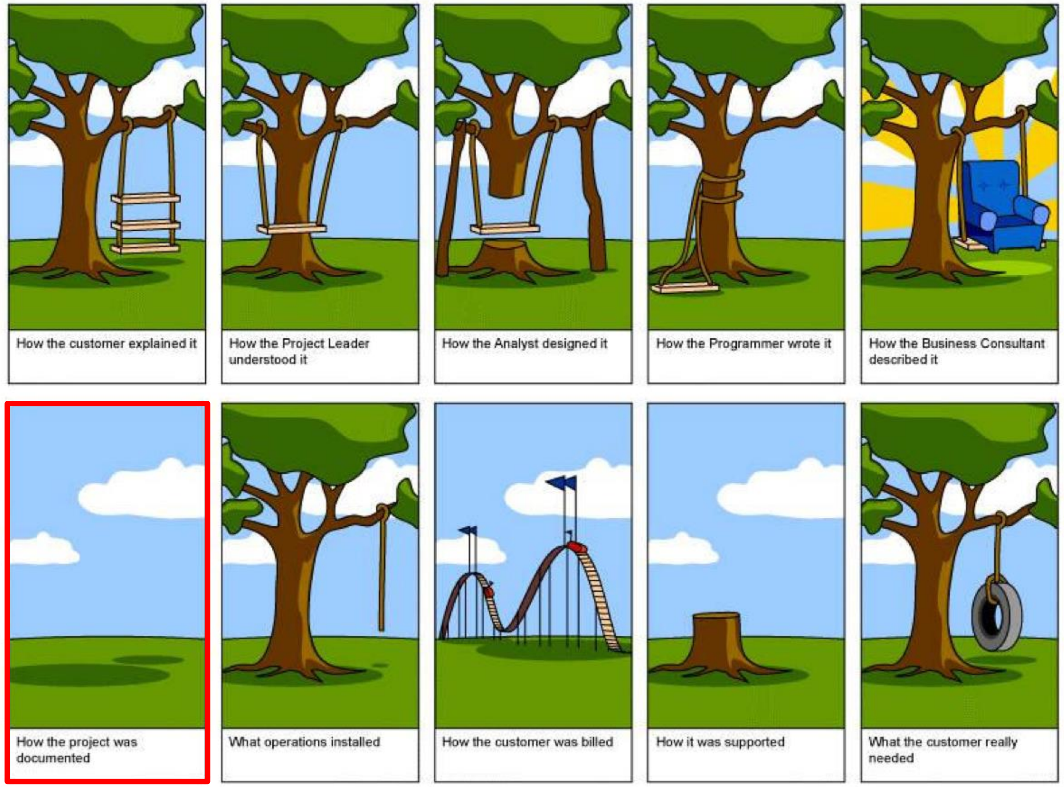
How Software Design and Engineering really works..


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Keep the problem as small as possible!



How Software Design and Engineering really works..


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Documentation

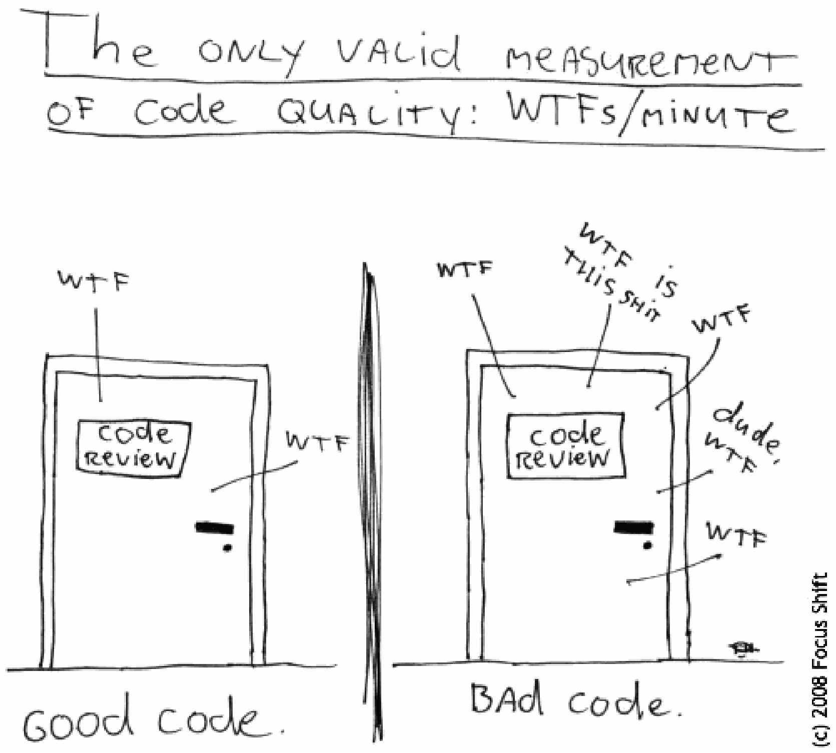
Technische Universität München

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Documentation for developers

This includes:

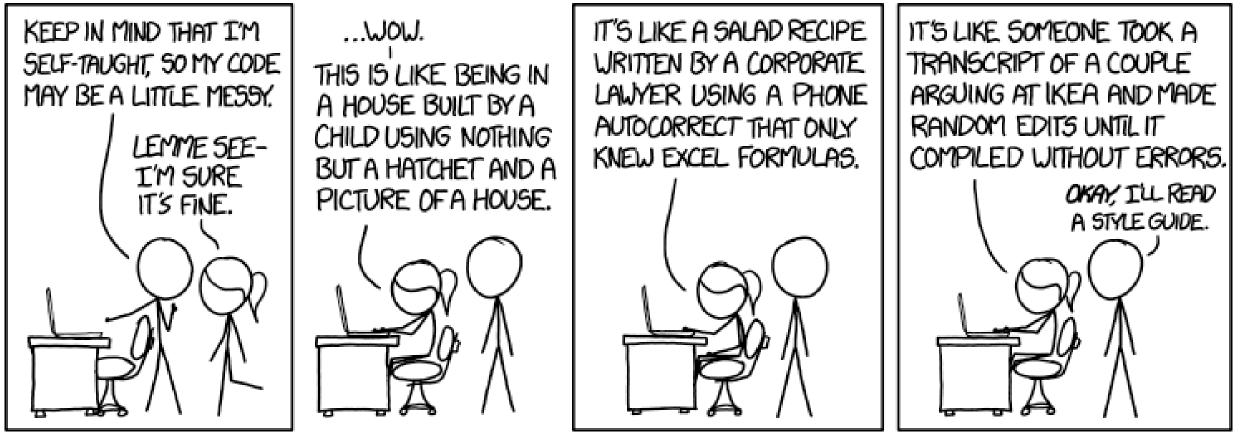
- Your customers
- Your team
- Yourself!



Documentation for developers – Code style

- Code is written once, but read many more times

- Don’t be lazy:
  - Good variable names
  - Refactor code
  - Keep modular and generic


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE– I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.

Documentation for developers – Comments

- No trivial comments
- Explain:
  - Assumptions
  - Corner cases
  - Non-trivial use of language features

BAD:

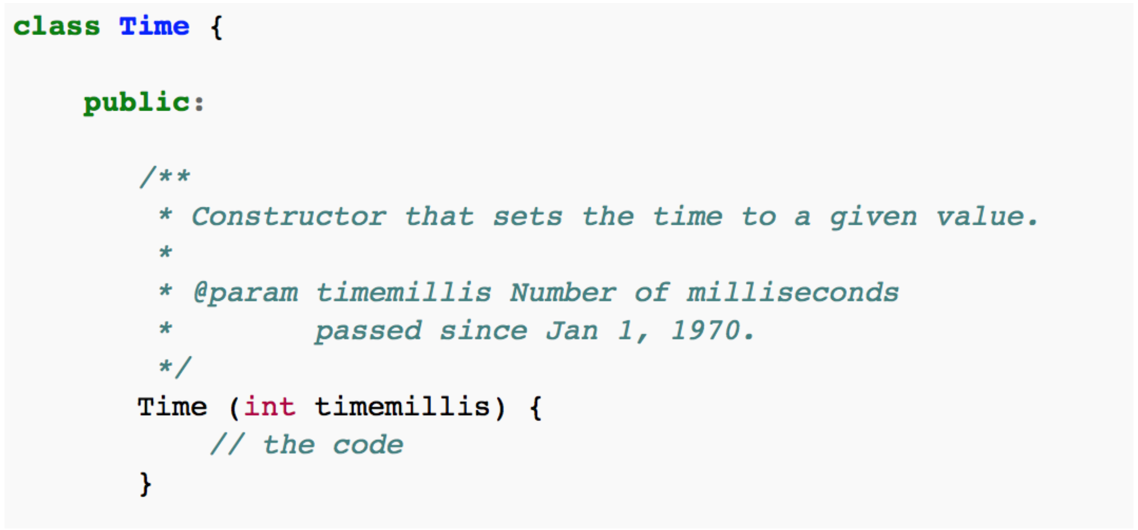
```
//Apply style.
apply(style);
```

GOOD:

```
// Unlike the others, this image needs to be drawn in the user-requested style
apply(style);
```

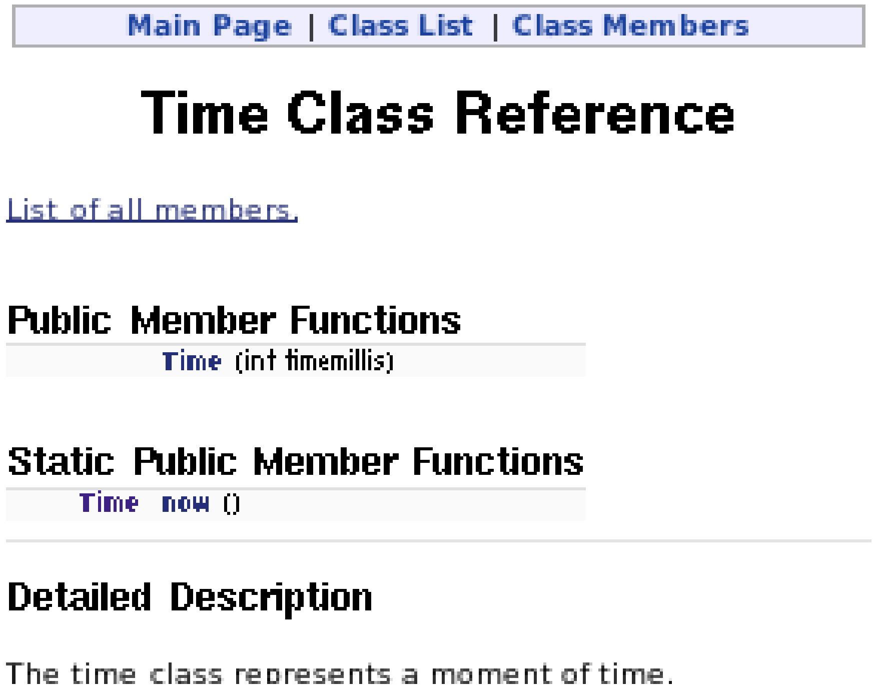
Documentation for developers – Doxygen

- Creates static docs from comments
- Close to source code, so USUALLY less out-of-date
- Useful only with non-trivial content



```
class Time {

    public:

      /**
       * Constructor that sets the time to a given value.
       *
       * @param timemillis Number of milliseconds
       *        passed since Jan 1, 1970.
       */
      Time (int timemillis) {
          // the code
      }
```

Documentation for developers – Doxygen



Main Page | Class List | Class Members

Time Class Reference

List of all members.

Public Member Functions

Time (int timemillis)

Static Public Member Functions

Time now ()

Detailed Description

The time class represents a moment of time.

Documentation for users

- Users as seen by developers:

- Usually the cause is bad documentation!
- You make a lot of assumptions that are clear in your head, but not to a new user

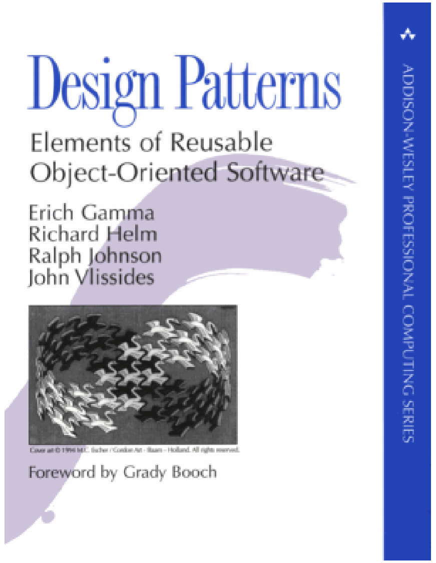
Design Patterns (and anti-Patterns)

Technische Universität München

Johns Hopkins Whiting School of Engineering

Design Patterns

- Reusable code structures
- Solve common problems
- Proven to work, common vocabulary
- Mostly created to work around rigid Object-Oriented type systems
- BUT: focus on the problem rather than where to stuff them in your program!


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Some design Patterns

- Singleton: class with only one instance in whole program
- Abstract factory: allows to create an instance of several families of classes
- Observer: way of notifying change to a number of classes
- Decorator: add functionality to class without inheriting
- Facade: single class that represents an entire subsystem

SourceMaking, “Design Patterns,” [Online] Available: https://sourcemaking.com/design_patterns

Design anti-Patterns

- Too many classes

- Functions too long

```
img_filter = ImageFilter()
img_filter.set_image(img)
img_filter.set_radius(2.5)
filtered_img = img_filter.get_output()
```

→

```
filtered_img = filter_img(img, radius=2.5)
```

Design anti-Patterns

- Too many classes
- Functions too long
- Mixed functionality
- Reinventing the wheel
- Premature optimization

Testing

Technische Universität München

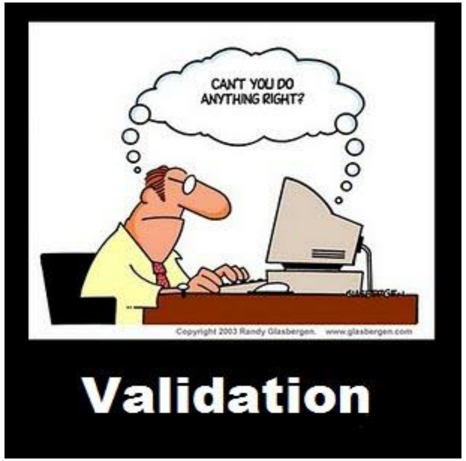
+

Johns Hopkins
Whiting School
of Engineering

Testing – Definitions

- Verification and Validation (V&V)
  - Verification: The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of the phase [IEEE-STD-610]
  - Validation: The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements [IEEE-STD-610]

UNIVERSAL
HUGE, PUBLICLY FUNDED PROJECT CHECKLIST
COME UP WITH IDEA.
GET PEOPLE REALLY EXCITED ABOUT IT.
FIND SOME EXPERT TO SAY IT WILL BE FINANCIALLY FEASIBLE.
BUILD IT.
WATCH IT STRUGGLE, NOT LIVE UP TO PROJECTIONS AND COST TAXPAYERS WAY MORE THAN THEY WERE TOLD.
nealo.com
Verification


CAN'T YOU DO ANYTHING RIGHT?
Copyright 2003 Randy Glasbergen. www.glasbergen.com
Validation

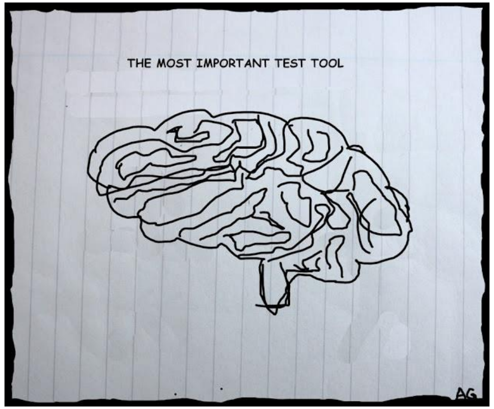
Testing – Definitions

Criteria	Verification	Validation
Definition	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
Objective	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.
Question	Are we building the product *right*?	Are we building the *right* product?
Evaluation Items	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.
Activities	•Reviews •Walkthroughs •Inspections	•Testing

Test types

- Runtime Test: Sanity check for invalid program states during runtime
- Test Run: Developer runs the software and looks for obvious errors
- Systematic Test: Carefully chosen test data, comparison with expected results
- Regression Test: Extended and automated systematic test, run repeatedly (e.g. after every commit), test results are documented
- Performance Test: Testing performance of the software (runtime, memory usage, ...)

Testing may be a pain in the neck, but with the right combination of the above test types you get a good cost-return value



Test levels

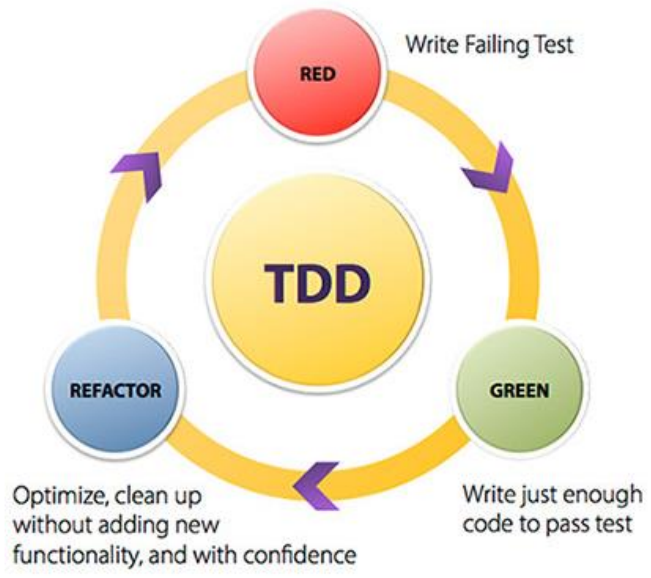
- Unit Test: Checks a single piece of code (e.g. class) in isolation
- Integration Test: Verifies the interfaces between components
- System Test: Checks that the whole software meets the requirements
- Operational Acceptance Test: Put the software to test with real end users and in realistic conditions

Unit test

```python
import unittest

def fun(x):
    return x + 1

class MyTest(unittest.TestCase):
    def test(self):
        self.assertEqual(fun(3), 4)
```

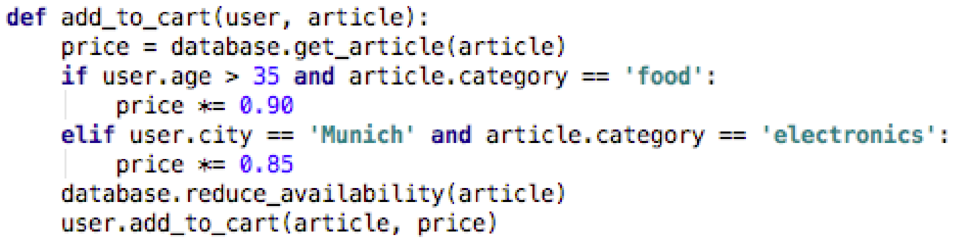
Test Driven Development

- Write tests first, then develop until pass
- Pros:
  - Help focusing on objectives
  - Think about corner cases
  - More rewarding experience
  - More confident about later changes


Write Failing Test
RED
TDD
REFACTOR
GREEN
Optimize, clean up without adding new functionality, and with confidence
Write just enough code to pass test

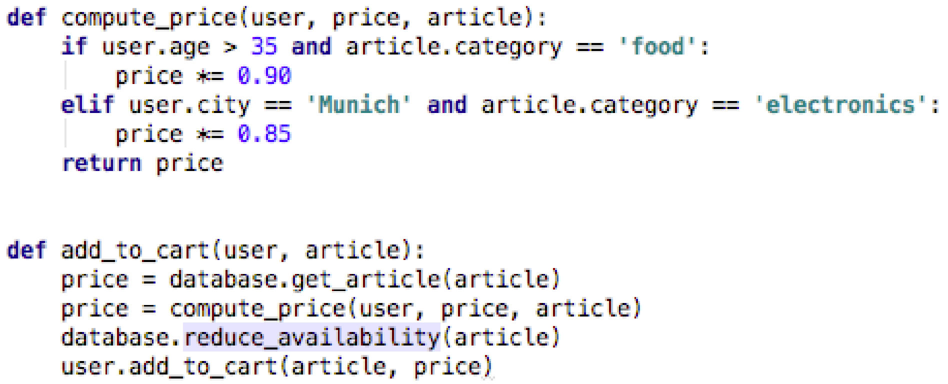
Testable code

- Keep functions small



```python
def add_to_cart(user, article):
    price = database.get_article(article)
    if user.age > 35 and article.category == 'food':
        price *= 0.90
    elif user.city == 'Munich' and article.category == 'electronics':
        price *= 0.85
    database.reduce_availability(article)
    user.add_to_cart(article, price)
```

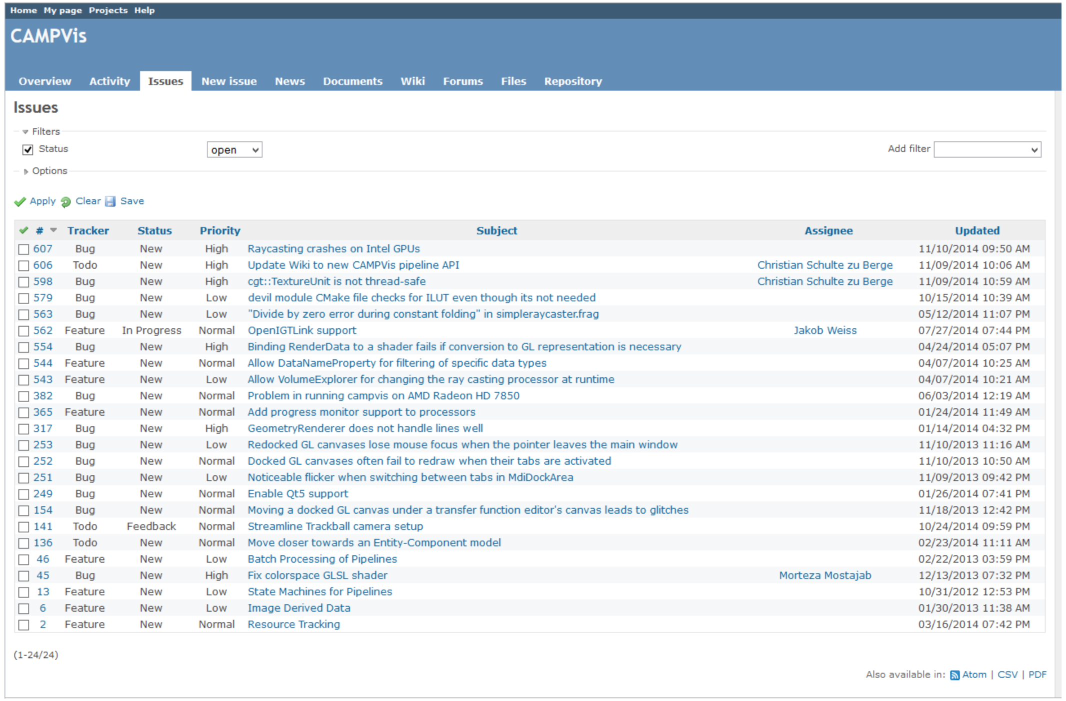
- Do not mix functionality



```python
def compute_price(user, price, article):
    if user.age > 35 and article.category == 'food':
        price *= 0.90
    elif user.city == 'Munich' and article.category == 'electronics':
        price *= 0.85
    return price


def add_to_cart(user, article):
    price = database.get_article(article)
    price = compute_price(user, price, article)
    database.reduce_availability(article)
    user.add_to_cart(article, price)
```

Bug tracker

- Help tracking defects present in software


Home My page Projects Help
CAMPVis
Overview Activity Issues New issue News Documents Wiki Forums Files Repository

Issues

Filters

[✓] Status open

Add filter

Options

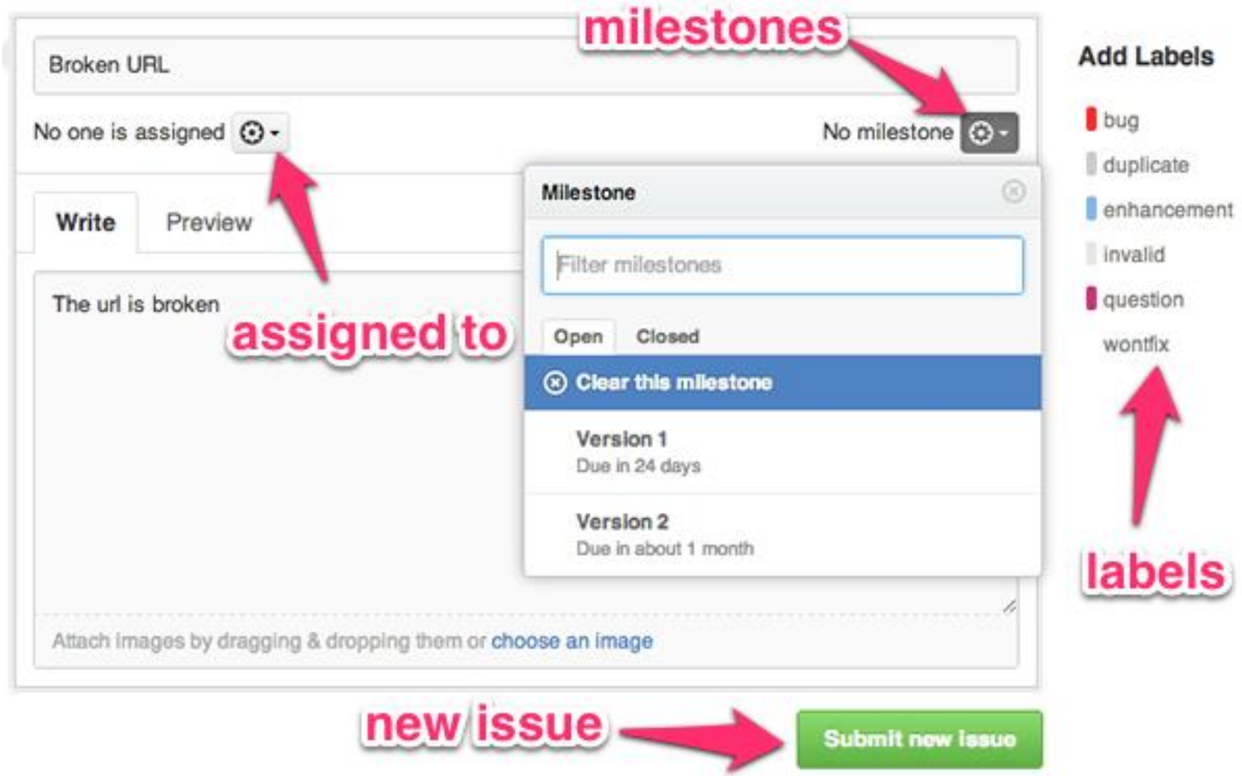
Apply Clear Save

#	Tracker	Status	Priority	Subject	Assignee	Updated
607	Bug	New	High	Raycasting crashes on Intel GPUs		11/10/2014 09:50 AM
606	Todo	New	High	Update Wiki to new CAMPVis pipeline API	Christian Schulte zu Berge	11/09/2014 10:06 AM
598	Bug	New	High	cgt::TextureUnit is not thread-safe	Christian Schulte zu Berge	11/09/2014 10:59 AM
579	Bug	New	Low	devil module CMake file checks for ILUT even though its not needed		10/15/2014 10:39 AM
563	Bug	New	Low	"Divide by zero error during constant folding" in simpleraycaster.frag		05/12/2014 11:07 PM
562	Feature	In Progress	Normal	OpenIGTLink support	Jakob Weiss	07/27/2014 07:44 PM
554	Bug	New	High	Binding RenderData to a shader fails if conversion to GL representation is necessary		04/24/2014 05:07 PM
544	Feature	New	Normal	Allow DataNameProperty for filtering of specific data types		04/07/2014 10:25 AM
543	Feature	New	Low	Allow VolumeExplorer for changing the ray casting processor at runtime		04/07/2014 10:21 AM
382	Bug	New	Normal	Problem in running campvis on AMD Radeon HD 7850		06/03/2014 12:19 AM
365	Feature	New	Normal	Add progress monitor support to processors		01/24/2014 11:49 AM
317	Bug	New	High	GeometryRenderer does not handle lines well		01/14/2014 04:32 PM
253	Bug	New	Low	Redocked GL canvases lose mouse focus when the pointer leaves the main window		11/10/2013 11:16 AM
252	Bug	New	Normal	Docked GL canvases often fail to redraw when their tabs are activated		11/10/2013 10:50 AM
251	Bug	New	Low	Noticeable flicker when switching between tabs in MdiDockArea		11/09/2013 09:42 PM
249	Bug	New	Normal	Enable Qt5 support		01/26/2014 07:41 PM
154	Bug	New	Normal	Moving a docked GL canvas under a transfer function editor's canvas leads to glitches		11/18/2013 12:42 PM
141	Todo	Feedback	Normal	Streamline Trackball camera setup		10/24/2014 09:59 PM
136	Todo	New	Normal	Move closer towards an Entity-Component model		02/23/2014 11:11 AM
46	Feature	New	Low	Batch Processing of Pipelines		02/22/2013 03:59 PM
45	Bug	New	High	Fix colorspace GLSL shader	Morteza Mostajab	12/13/2013 07:32 PM
13	Feature	New	Low	State Machines for Pipelines		10/31/2012 12:53 PM
6	Feature	New	Low	Image Derived Data		01/30/2013 11:38 AM
2	Feature	New	Normal	Resource Tracking		03/16/2014 07:42 PM

(1-24/24)

Also available in: Atom | CSV | PDF

Bug tracker


Broken URL

No one is assigned

No milestone

Write | Preview

The url is broken

Attach images by dragging & dropping them or choose an image

Milestone

Filter milestones

Open | Closed

Clear this milestone

Version 1
Due in 24 days

Version 2
Due in about 1 month

Add Labels

- bug
- duplicate
- enhancement
- invalid
- question
- wontfix

Integration strategies

Technische Universität München

+

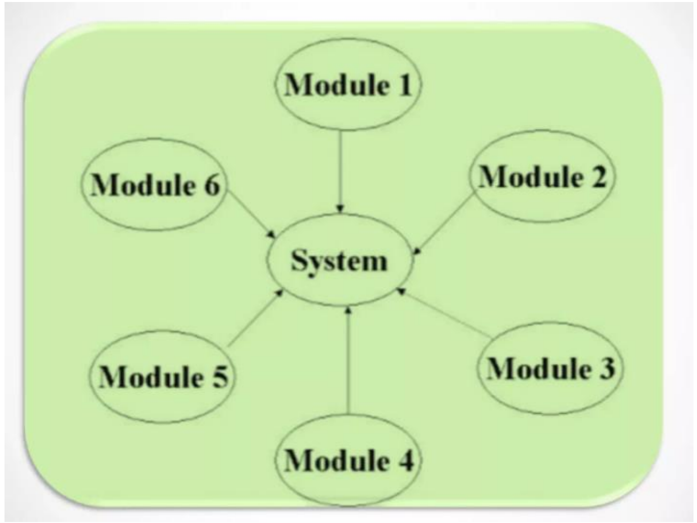
JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

The Big-Bang Integration Strategy

- *Unordered implementation of the components / all components implemented at the same time*

- Problems
  - Errors are very hard to locate: Which component is the cause?
  - Design errors (errors in interfaces) not distinguishable from implementation errors

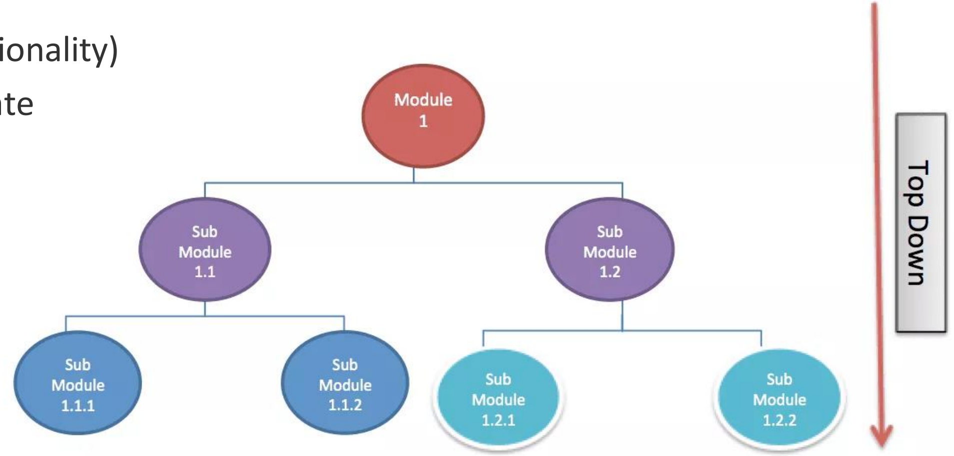
- Always prefer incremental integration strategy



Top-Down Integration Strategy

- *Start with the components from the top-most layer (e.g. GUI). Incrementally add layers further down*

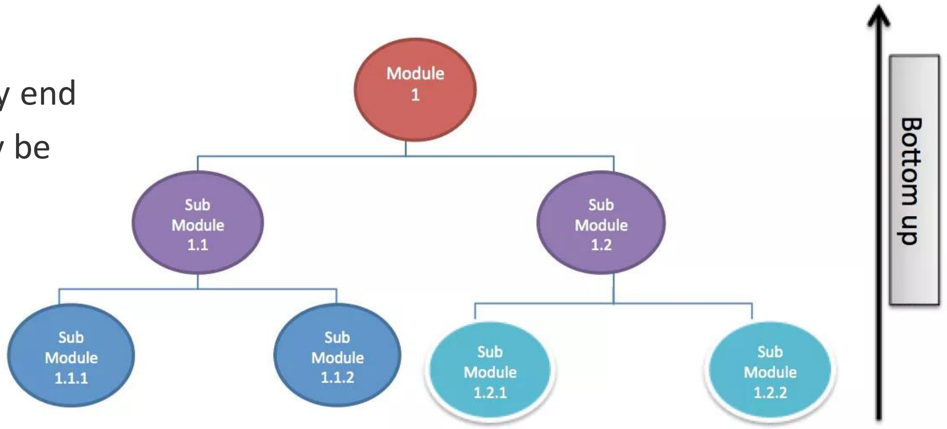
- Pros/Cons
  - Early prototype available (with limited functionality)
  - Design errors can be detected in an early state
  - Many stubs required → cumbersome
  - No functionality until a very late stage



Bottom-Up Integration Strategy

- *Start with the components from the bottom-most layer (e.g. entity classes). Incrementally add upper layers.*

- Pros/Cons
  - No stubs required
  - Functionality available in early stages
  - Nothing to show to customers until the very end
  - Errors may be expensive, because they may be found late and solving them might require cumbersome changes



Version control

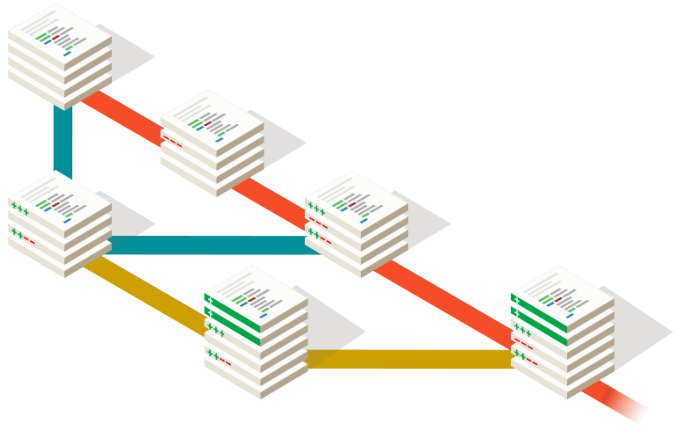
Technische Universität München

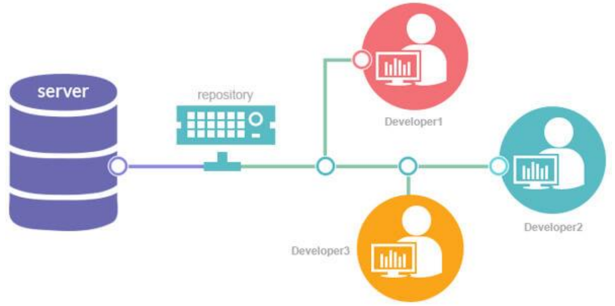
+

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

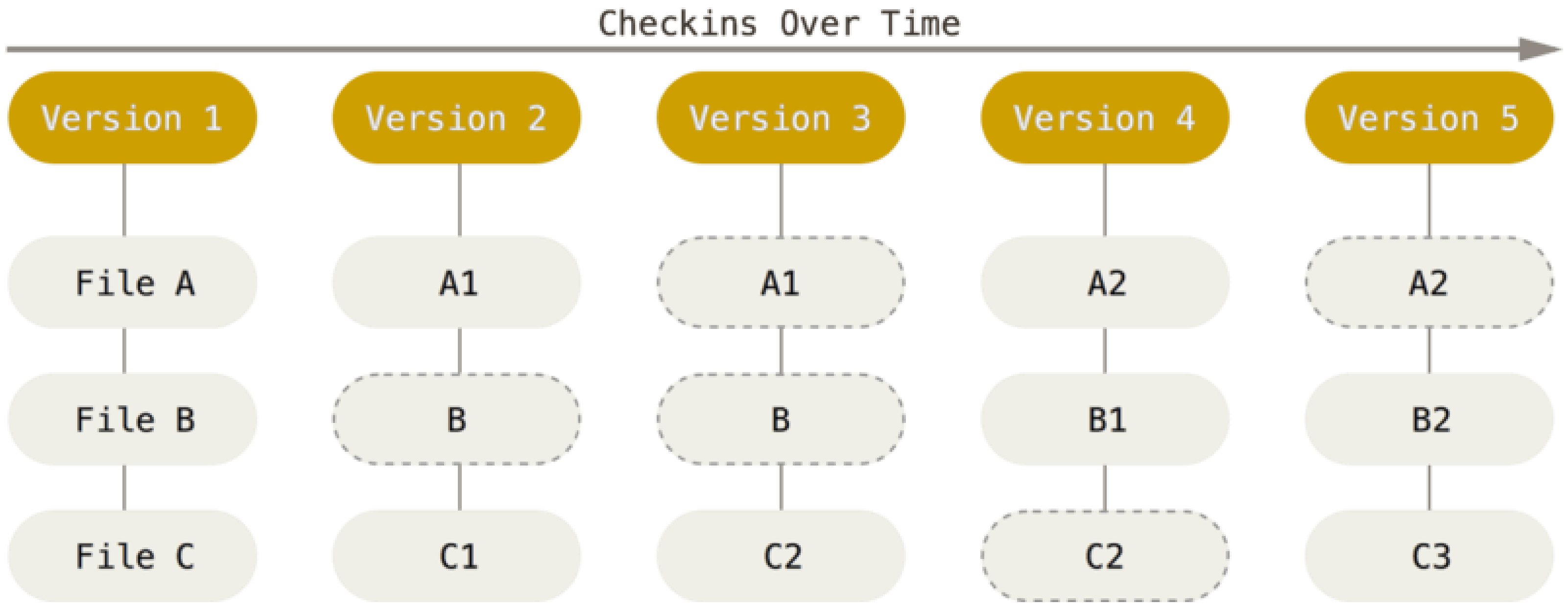
Version Control Systems

- Keep a history of changes to code
- Share code with others
- Integrate changes from others
- Manage concurrent versions

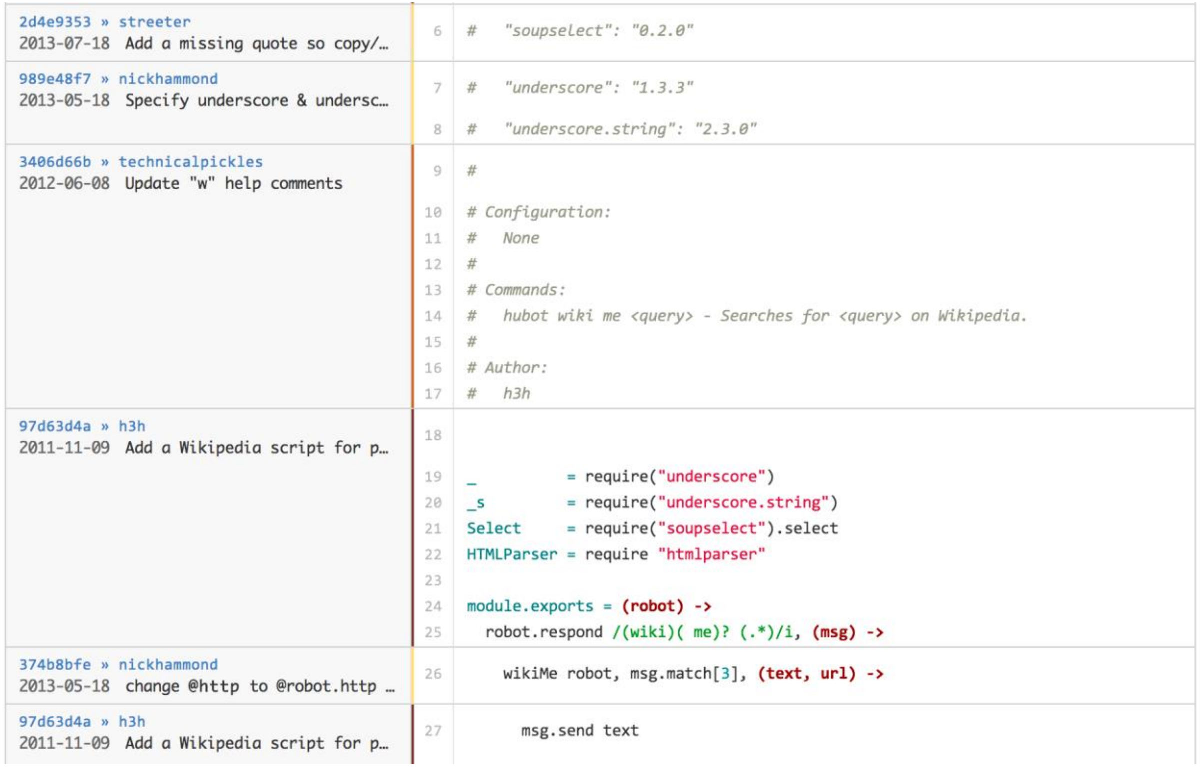




Version history

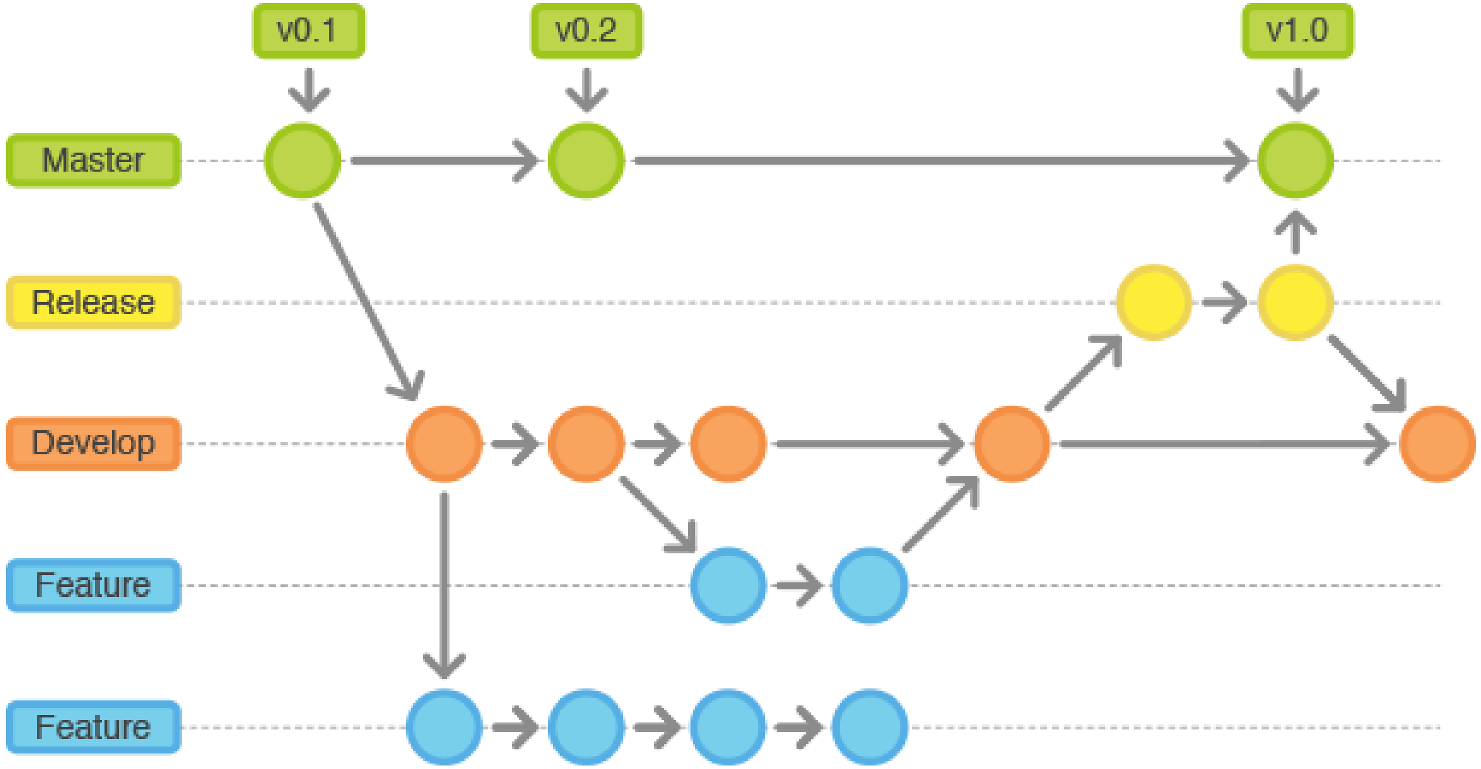


Changes history

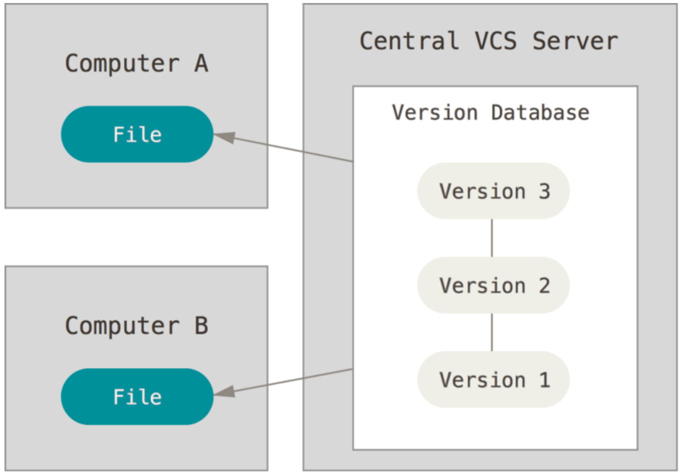


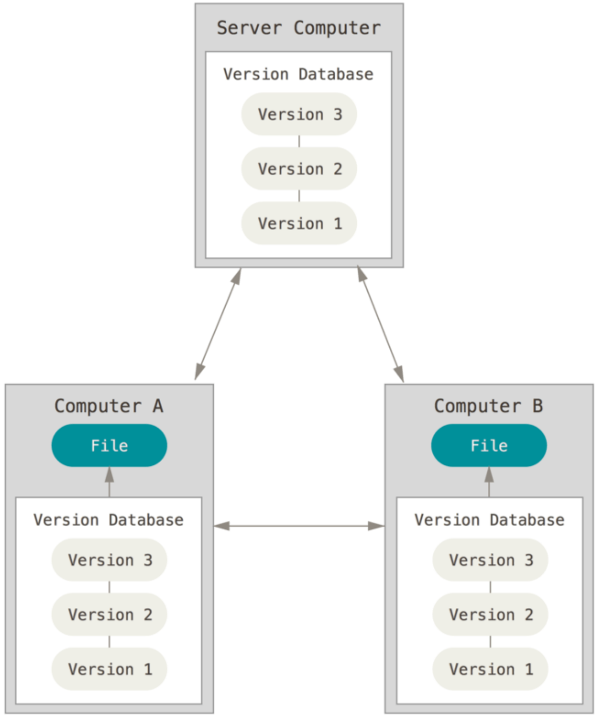
```
2d4e9353 » streeter
2013-07-18  Add a missing quote so copy/…
#   "soupselect": "0.2.0"

989e48f7 » nickhammond
2013-05-18  Specify underscore & undersc…
#   "underscore": "1.3.3"
#   "underscore.string": "2.3.0"

3406d66b » technicalpickles
2012-06-08  Update "w" help comments
#
# Configuration:
#   None
#
# Commands:
#   hubot wiki me <query> - Searches for <query> on Wikipedia.
#
# Author:
#   h3h

97d63d4a » h3h
2011-11-09  Add a Wikipedia script for p…

_          = require("underscore")
_s         = require("underscore.string")
Select     = require("soupselect").select
HTMLParser = require "htmlparser"

module.exports = (robot) ->
  robot.respond /(wiki)( me)? (.*)/i, (msg) ->

374b8bfe » nickhammond
2013-05-18  change @http to @robot.http …
    wikiMe robot, msg.match[3], (text, url) ->

97d63d4a » h3h
2011-11-09  Add a Wikipedia script for p…
      msg.send text
```

Branches

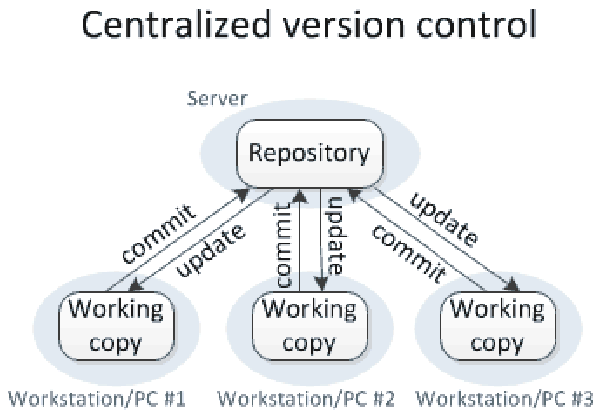


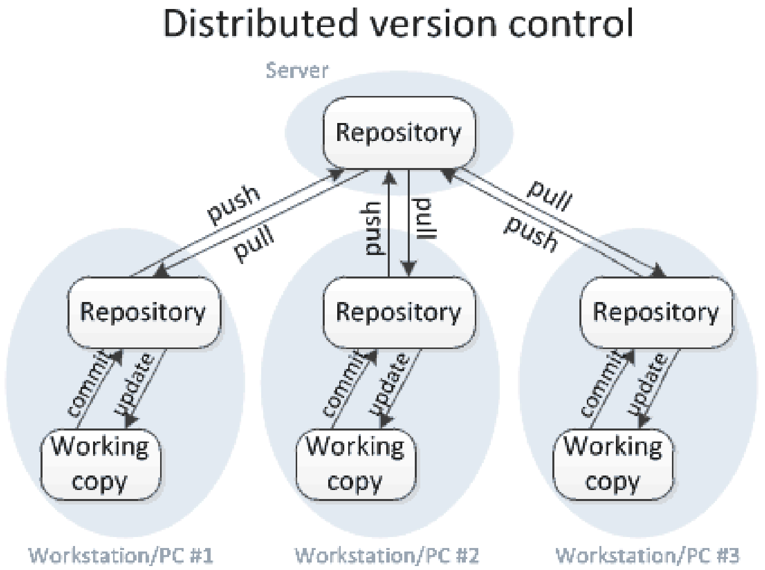
Centralized vs Distributed Version Control Systems





Centralized vs Distributed Version Control Systems

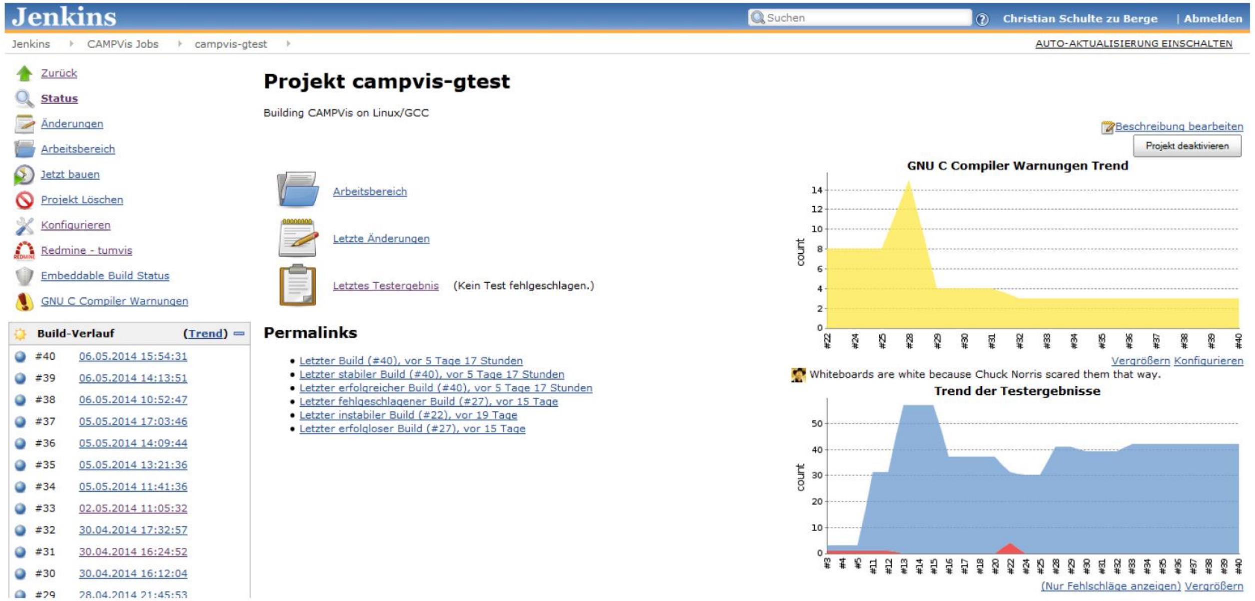




Software Configuration Management Guide, “Centralized vs Distributed Version Control Systesm,”
[Online] Available: https://scmquest.com/centralized-vs-distributed-version-control-systems/

Continuous Integration

- Compile automatically on every change uploaded to VCS



Jenkins | Suchen | Christian Schulte zu Berge | Abmelden

Jenkins › CAMPVis Jobs › campvis-gtest › AUTO-AKTUALISIERUNG EINSCHALTEN

- Zurück
- Status
- Änderungen
- Arbeitsbereich
- Jetzt bauen
- Projekt Löschen
- Konfigurieren
- Redmine - tumvis
- Embeddable Build Status
- GNU C Compiler Warnungen

Build-Verlauf (Trend)

#40	06.05.2014 15:54:31
#39	06.05.2014 14:13:51
#38	06.05.2014 10:52:47
#37	05.05.2014 17:03:46
#36	05.05.2014 14:09:44
#35	05.05.2014 13:21:36
#34	05.05.2014 11:41:36
#33	02.05.2014 11:05:32
#32	30.04.2014 17:32:57
#31	30.04.2014 16:24:52
#30	30.04.2014 16:12:04
#29	28.04.2014 21:45:53

Projekt campvis-gtest

Building CAMPVis on Linux/GCC

Beschreibung bearbeiten | Projekt deaktivieren

- Arbeitsbereich
- Letzte Änderungen
- Letztes Testergebnis (Kein Test fehlgeschlagen.)

Permalinks

- Letzter Build (#40), vor 5 Tage 17 Stunden
- Letzter stabiler Build (#40), vor 5 Tage 17 Stunden
- Letzter erfolgreicher Build (#40), vor 5 Tage 17 Stunden
- Letzter fehlgeschlagener Build (#27), vor 15 Tage
- Letzter instabiler Build (#22), vor 19 Tage
- Letzter erfolgloser Build (#27), vor 15 Tage

Whiteboards are white because Chuck Norris scared them that way.

Thank you

Happy coding

Ardit Ramadani, M.Sc.
Research Assistant

Deutsches Herzzentrum München des Freistaates Bayern
Klinik an der Technischen Universität München
Lazarettstr. 36
80636 München

Technische Universität München
Fakultät für Informatik - I16
Chair of Computer Aided Medical Procedures and Augmented Reality
Boltzmannstr. 3
85748 Garching bei München

https://www.in.tum.de/campar/members/ardit-ramadani/
ardit.ramadani@tum.de
ramadani@dhm.mhn.de

Technische Universität München + JOHNS HOPKINS WHITING SCHOOL *of* ENGINEERING