Software Development

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

AMOS E01

Software Development Team (Recap)

- The **software development team**
  - Holds **overall responsibility** for **delivering working software**
    - That provides the **features** the **team committed to delivering**

Primary Role Responsibilities (Recap)

- Engineering Management
  - Who?
    - By when?
- Software Development
  - How?
    - How fast?
- Quality Assurance
  - Releasable?
    - Good enough?

Sprint Planning (Practices)

- Break Down Features into Tasks
  - Responsible: Developers
  - Artifacts: Feature, task board, tasks
  - Collaborators: Developers
- Track Feature Implementation
  - Responsible: Developers
  - Artifacts: Sprint backlog, task board
  - Collaborators: Developers

KB AMSE WS 2015

Display another project | Logout (Dirk Riehle)

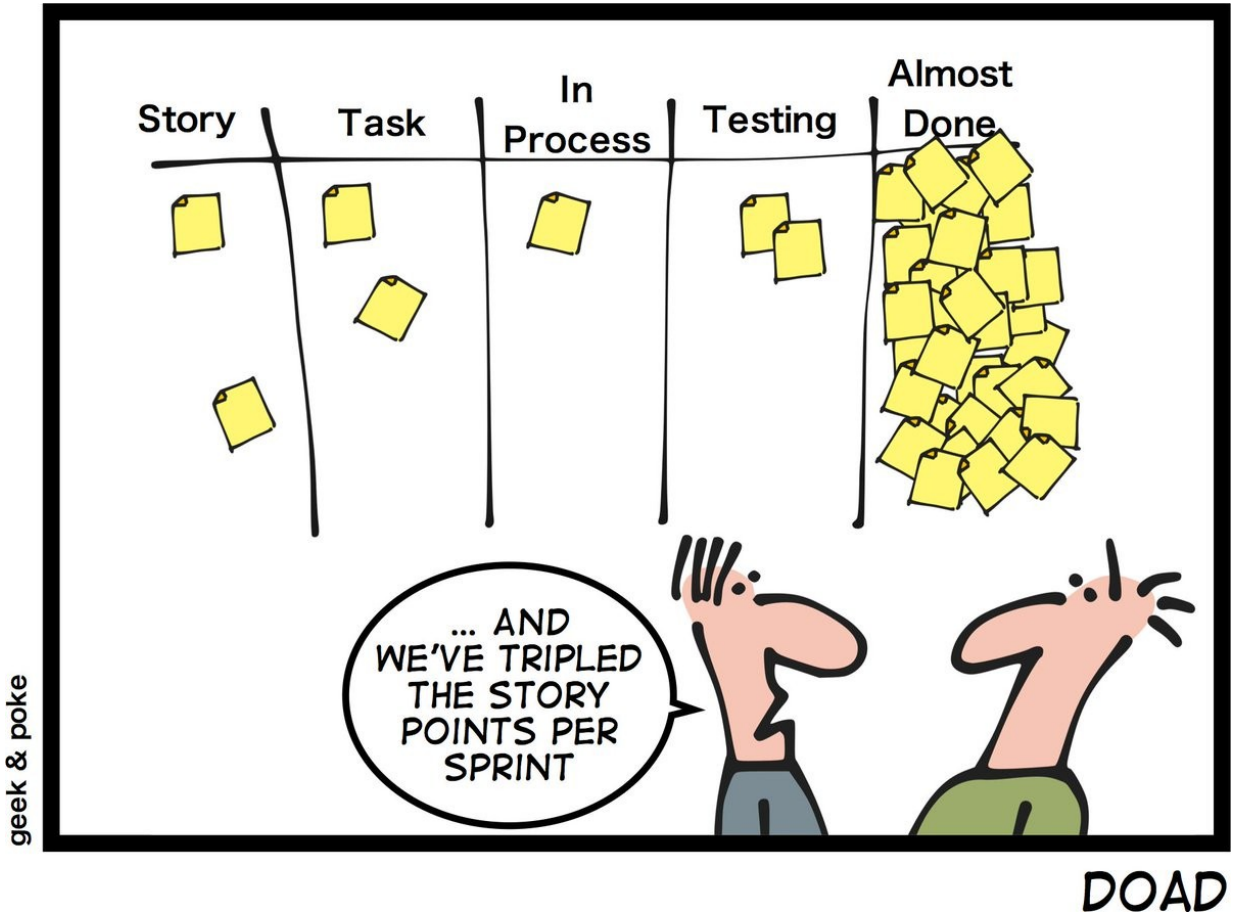
Actions | Board | Calendar | List | Gantt | status:open | Filters | Users | Categories

	Backlog (6)	Ready (46)	Work in progress (0)	Done (0)
Default swimlane				
New WOM Persistence	Backlog (8)	Ready (3)	Work in progress (1)	Done (0)
	Backlog (5)	Ready (9)	Work in progress (0)	Done (3)
Multi-Project Support (17)	#133 Nobody assigned MultiProject: Get rid of deprecated methods and method calls #341 Nobody assigned Add feature to hide resources from certain commits #368 Nobody assigned HEAD pointer must be stored as foreign key to branch ref or commit id #370 Nobody assigned Think about NoSuchCommitException #403 Nobody assigned List<Project> getProjectsByProjectOwner(String projectOwnerName); should return list of names	#26 Nobody assigned Fully implement and test PageableIterator for users and projects #296 Hannes Dohrn Let WRI implement Comprarable intenrface #388 Nobody assigned Implement list resources pagination when accessed via web api #391 Nobody assigned Add boolean switch to getProjects method that let's you decide whether you want to see deleted projects or not #185 Jonas Gröger Implement project specific roles #429 Nobody assigned TODO: We should probably mirror the resource paths from the web UI		#151 Hannes Dohrn Draft design: Reimplement Sweble UI in REST+JS #176 Hannes Dohrn RFC: REST Server + Web Server 2 4 #187 Hannes Dohrn UI design proposal 1
	Backlog (3)	Ready (0)	Work in progress (0)	Done (0)
Future (3)	#10 Nobody assigned Implement REST API for transaction interface #11 Nobody assigned Wikipedia Integration			

Task Board (Artifact)

- Task board
  - Visualizes the progress towards finishing the current sprint
  - Shows for each feature the progress of implementation
  - May break down work into hours

Definition of Almost Done



Day Planning (Practices)

- Perform Daily Scrum
  - Responsible: Scrum master
  - Artifacts: Impediment backlog
  - Collaborators: Developers

Daily Scrum

- Daily Scrum
  - Is a daily status meeting to sync on problems and upcoming work
  - Is to be kept as short as possible (a.k.a. daily stand-up meeting)
- Other properties
  - “Pigs” are mandatory, “chicken” are optional, only “pigs” may speak
  - Provides only updates, everyone may only speak once
  - No discussions allowed, any discussions are taken off-line
  - Scrum master to follow-up on problems
- Questions asked
  - What did you do yesterday?
  - What will you be doing today?
  - What obstacles are in your way?

Plank Meetings

Collective Code Ownership (Practice)

- Own code collectively
  - Responsible: Development team
  - Artifacts: Source code
  - Collaborators: Developers

Collective Code Ownership

- Definition and purpose
  - Provide full read and write access to each developer of the collective
  - Is to instill a feeling of overall responsibility for the code base
  - The opposite of collective code ownership is individual code ownership
- Other properties
  - Every member of the collective
    - Cares about the architecture
    - Cares about clean code
    - Ensures high quality
  - Every single author can
    - Implement a feature end-to-end
    - Finish a refactoring
    - Fix a bug

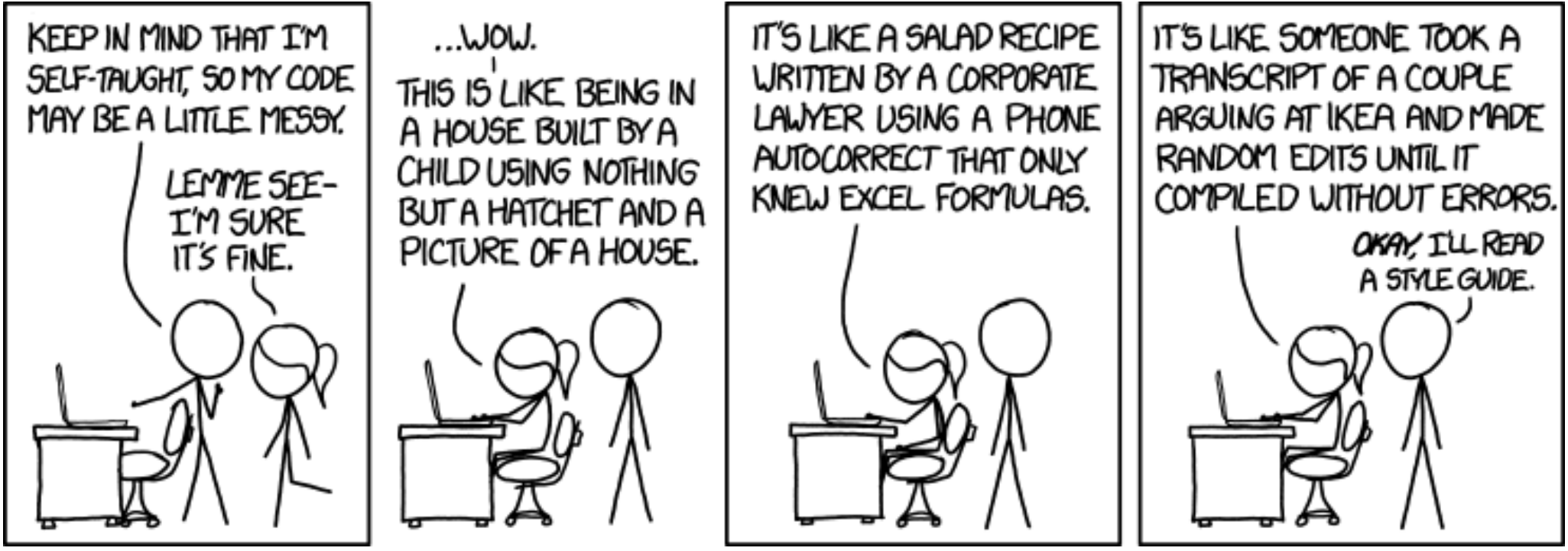
Programming Standard (Practice)

- Use programming standard
  - Responsible: Development team
  - Artifacts: Programming standard, source code
  - Collaborators: Developers

Programming Standard

- Definition and purpose
  - Is a set of rules and conventions that determines naming, formating and structuring of source code and related artifacts
  - Is used to ensure that every developer can read and modify every other developer's code as easily as possible
- Other properties
  - Ease reading source code
  - Ease navigating code
  - Should be mandatory

xkcd on Programming Standards



Example Java Programming Standard 1 / 4

3.1.1 Beginning Comments

All source files should begin with a c-style comment that lists the class name, version information, date, and copyright notice:

```
/*
 * Classname
 *
 * Version information
 *
 * Date
 *
 * Copyright notice
 */

...
```

Example Java Programming Standard 2 / 4

4.2 Wrapping Lines

When an expression will not fit on a single line, break it according to these general principles:

- Break after a comma.
- Break before an operator.
- Prefer higher-level breaks to lower-level breaks.

...

Example Java Programming Standard 3 / 4

6.3 Placement

Put declarations only at the beginning of blocks. (A block is any code surrounded by curly braces "{" and "}".) Don't wait to declare variables until their first use; it can confuse the unwary programmer and hamper code portability within the scope.

```
void myMethod() {
    int int1 = 0; // beginning of method block
    if (condition) {
        int int2 = 0; // beginning of "if" block
        ...
    }
}

...
```

Example Java Programming Standard 4 / 4

```
7.1 Simple Statements

Each line should contain at most one statement. Example:

argv++; // Correct
argc--; // Correct
argv++; argc--; // AVOID!

...
```

Categories and Examples of Method Types

Query Method	Mutation Method	Helper Method
get method (getter)	set method (setter)	factory method
boolean query method	command method	cloning method
comparison method	initialization method	assertion method
conversion method	finalization method	logging method
...	...	...

Get Method (Query Method)

Definition	A get method is a query method that returns a (logical) field of the object.
Also known as	Getter
JDK example	Class Object#getClass() Object Enumeration#nextElement()
Name example	String getComponent(int) Iterable<String> getComponentIterator()
Prefixes	get
Naming	After the prefix, the name of the field being queried follows.

Assertion Method (Helper Method)

Definition	An assertion method is a helper method that tests a condition. If the condition holds, it returns quietly. If it does not, an exception is thrown.
Also known as	-
JDK example	void AccessControlContext#checkPermission(Permission) throws AccessControlException
Name example	void assertIsValidIndex(int) throws InvalidIndexException
Prefixes	assert, check, test
Naming	After the prefix, the condition being checked follows.

Conventions / Patterns Beyond Formatting

- Naming conventions
  - Attributes, methods
  - Classes, packages
  - ...
- Design conventions
  - Collaborations
  - Modules, classes
  - ...
- Package structures
  - ...
- **See our Advanced Design and Programming (ADAP) course**

Quiz: Software Development

1. A file may have many authors. Should the names of these authors be listed in the file's header?
   - Yes
   - No

Build Process

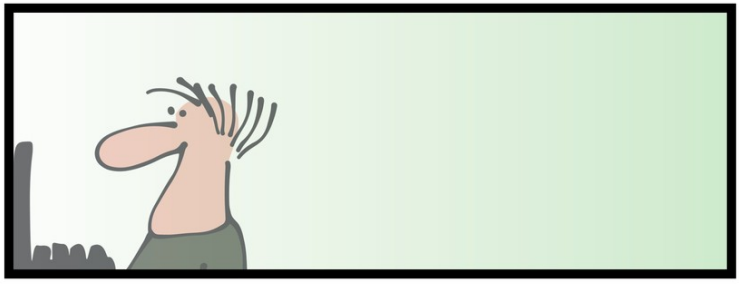
- Definition
  - Is the defined and (ideally automated) process of deriving an installable product from its source artifacts
- Purpose
  - Defines a standard environment
  - Provides developers with setup
  - Defines clear commit rules
  - Manages test data etc.

Developer Responsibilities

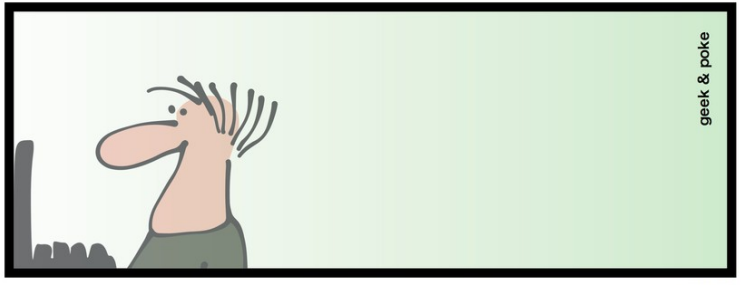
- Only work in defined standardized environment
  - This ensures no subtle differences to central build process
- Only commit / push code that compiles and works
  - “Breaking the build” through non-compiling code hurts team
- Only commit / push code that passes all the tests
  - Failing tests quickly degenerate system (hard to catch-up)

DEVELOPMENT CYCLE

FRIDAY EVENING EDITION



COMMIT



PUSH

RUN

Build Asset Management

- Management by hand
  - Libraries are curated by hand
- Automated management
  - Libraries are pulled in automatically
- Bill of materials
  - Possibly generated automatically

Quiz: Supply Chain Effects

- Last week you delivered your product release to your client. The phone rings and an angry client is on the line, complaining about missing their schedule. What might have gone wrong?
  1. The client's development team objected
  2. The client's quality assurance unit objected
  3. The client's legal team objected
  4. All of the above

Review / Summary of Session

- Development planning
  - Stories vs. tasks
  - Daily scrum
- Source code and coding
  - Collective code ownership
  - Programming guidelines
  - Other conventions
- Build process

Thank you! Questions?

dirk.riehle@fau.de – http://osr.cs.fau.de

dirk@riehle.org – http://dirkriehle.com – @dirkriehle

Credits and License

- Original version
  - © 2012-2019 Dirk Riehle, some rights reserved
  - Licensed under Creative Commons Attribution 4.0 International License
- Contributions
  - ...

Software Development

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

AMOS E01

It is Friedrich-Alexander University Erlangen-Nürnberg – FAU, in short.

Corporate identity wants us to say “Friedrich-Alexander University”.

Software Development Team (Recap)

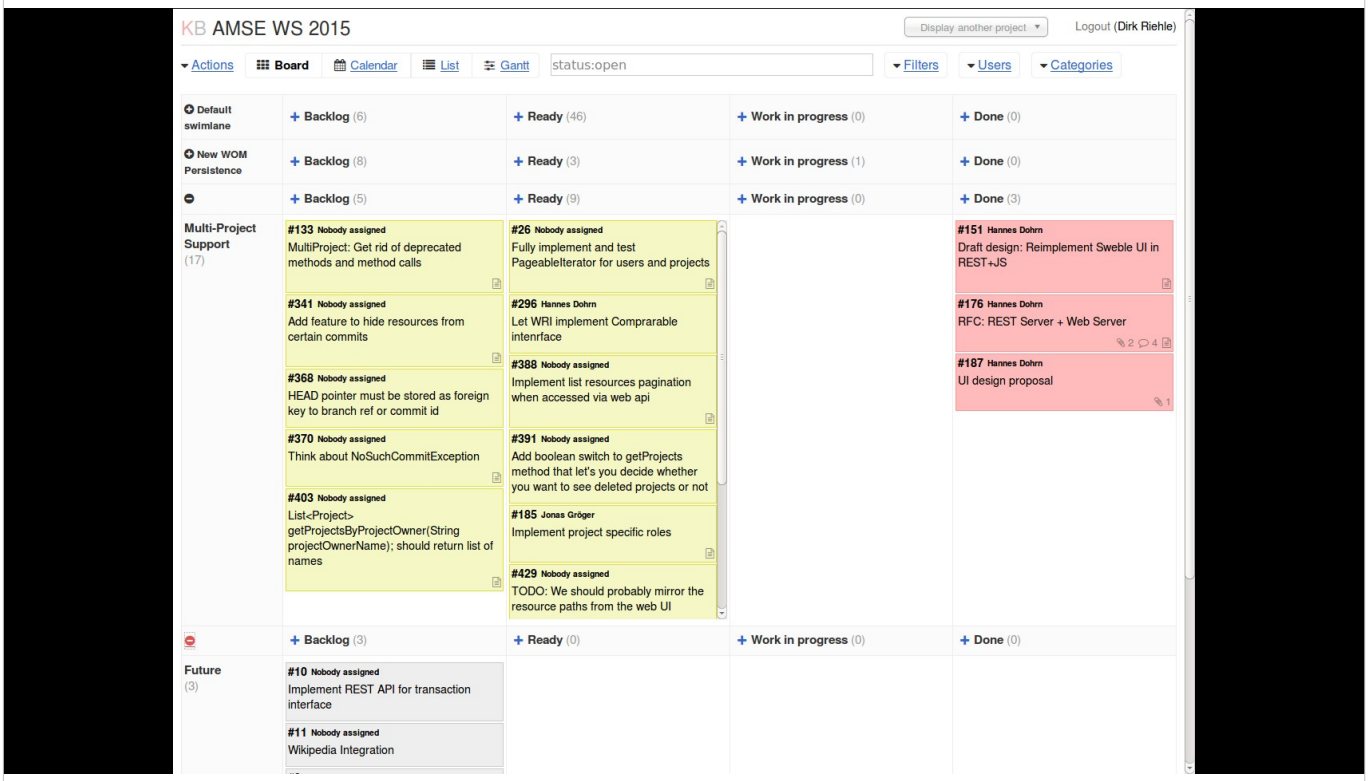
- The **software development team**
  - Holds **overall responsibility** for **delivering working software**
    - That provides the **features** the **team committed to delivering**

Primary Role Responsibilities (Recap)

- Engineering Management
  - Who?
    - By when?
- Software Development
  - How?
    - How fast?
- Quality Assurance
  - Releasable?
    - Good enough?

Sprint Planning (Practices)

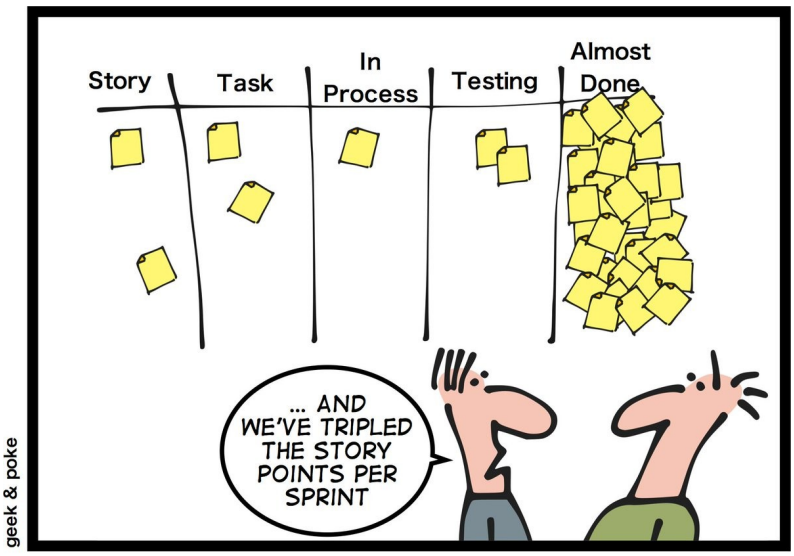
- Break Down Features into Tasks
  - Responsible: Developers
  - Artifacts: Feature, task board, tasks
  - Collaborators: Developers
- Track Feature Implementation
  - Responsible: Developers
  - Artifacts: Sprint backlog, task board
  - Collaborators: Developers


KB AMSE WS 2015
Display another project
Logout (Dirk Riehle)
Actions
Board
Calendar
List
Gantt
status:open
Filters
Users
Categories
Default swimlane
New WOM Persistence
Backlog (6)
Ready (46)
Work in progress (0)
Done (0)
Backlog (8)
Ready (3)
Work in progress (1)
Done (0)
Backlog (5)
Ready (9)
Work in progress (0)
Done (3)
Multi-Project Support (17)
#133 Nobody assigned
MultiProject: Get rid of deprecated methods and method calls
#341 Nobody assigned
Add feature to hide resources from certain commits
#368 Nobody assigned
HEAD pointer must be stored as foreign key to branch ref or commit id
#370 Nobody assigned
Think about NoSuchCommitException
#403 Nobody assigned
List<Project> getProjectsByProjectOwner(String projectOwnerName); should return list of names
#26 Nobody assigned
Fully implement and test PageableIterator for users and projects
#296 Hannes Dohrn
Let WRI implement Comprarable intenrface
#388 Nobody assigned
Implement list resources pagination when accessed via web api
#391 Nobody assigned
Add boolean switch to getProjects method that let's you decide whether you want to see deleted projects or not
#185 Jonas Gröger
Implement project specific roles
#429 Nobody assigned
TODO: We should probably mirror the resource paths from the web UI
#151 Hannes Dohrn
Draft design: Reimplement Sweble UI in REST+JS
#176 Hannes Dohrn
RFC: REST Server + Web Server
2 4
#187 Hannes Dohrn
UI design proposal
1
Backlog (3)
Ready (0)
Work in progress (0)
Done (0)
Future (3)
#10 Nobody assigned
Implement REST API for transaction interface
#11 Nobody assigned
Wikipedia Integration

Task Board (Artifact)

- Task board
  - Visualizes the progress towards finishing the current sprint
  - Shows for each feature the progress of implementation
  - May break down work into hours

Definition of Almost Done



DOAD

Day Planning (Practices)

- Perform Daily Scrum
  - Responsible: Scrum master
  - Artifacts: Impediment backlog
  - Collaborators: Developers

Plank Meetings

Collective Code Ownership (Practice)

- Own code collectively
  - Responsible: Development team
  - Artifacts: Source code
  - Collaborators: Developers

Collective Code Ownership

- Definition and purpose
  - Provide full read and write access to each developer of the collective
  - Is to instill a feeling of overall responsibility for the code base
  - The opposite of collective code ownership is individual code ownership
- Other properties
  - Every member of the collective
    - Cares about the architecture
    - Cares about clean code
    - Ensures high quality
  - Every single author can
    - Implement a feature end-to-end
    - Finish a refactoring
    - Fix a bug

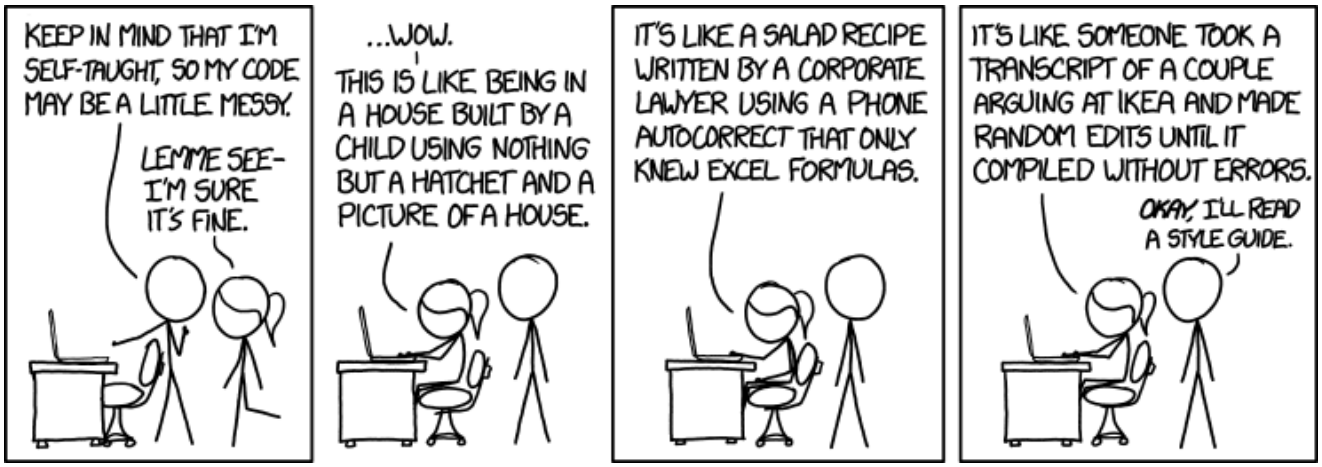
Programming Standard (Practice)

- Use programming standard
  - Responsible: Development team
  - Artifacts: Programming standard, source code
  - Collaborators: Developers

Programming Standard

- Definition and purpose
  - Is a set of rules and conventions that determines naming, formating and structuring of source code and related artifacts
  - Is used to ensure that every developer can read and modify every other developer's code as easily as possible
- Other properties
  - Ease reading source code
  - Ease navigating code
  - Should be mandatory

xkcd on Programming Standards



Example Java Programming Standard 1 / 4

3.1.1 Beginning Comments

All source files should begin with a c-style comment that lists the class name, version information, date, and copyright notice:

```
/*
 * Classname
 *
 * Version information
 *
 * Date
 *
 * Copyright notice
 */

...
```

Example Java Programming Standard 2 / 4

4.2 Wrapping Lines

When an expression will not fit on a single line, break it according to these general principles:

- Break after a comma.
- Break before an operator.
- Prefer higher-level breaks to lower-level breaks.

...

Example Java Programming Standard 3 / 4

6.3 Placement

Put declarations only at the beginning of blocks. (A block is any code surrounded by curly braces "{" and "}".) Don't wait to declare variables until their first use; it can confuse the unwary programmer and hamper code portability within the scope.

```
void myMethod() {
    int int1 = 0; // beginning of method block
    if (condition) {
        int int2 = 0; // beginning of "if" block
        ...
    }
}

...
```

Example Java Programming Standard 4 / 4

7.1 Simple Statements

Each line should contain at most one statement. Example:

```
argv++; // Correct
argc--; // Correct
argv++; argc--; // AVOID!

...
```

Categories and Examples of Method Types

Query Method	Mutation Method	Helper Method
get method (getter)	set method (setter)	factory method
boolean query method	command method	cloning method
comparison method	initialization method	assertion method
conversion method	finalization method	logging method
...	...	...

Get Method (Query Method)

Definition	A get method is a query method that returns a (logical) field of the object.
Also known as	Getter
JDK example	Class Object#getClass() Object Enumeration#nextElement()
Name example	String getComponent(int) Iterable<String> getComponentIterator()
Prefixes	get
Naming	After the prefix, the name of the field being queried follows.

Assertion Method (Helper Method)

Definition	An assertion method is a helper method that tests a condition. If the condition holds, it returns quietly. If it does not, an exception is thrown.
Also known as	-
JDK example	void AccessControlContext#checkPermission(Permission) throws AccessControlException
Name example	void assertIsValidIndex(int) throws InvalidIndexException
Prefixes	assert, check, test
Naming	After the prefix, the condition being checked follows.

Conventions / Patterns Beyond Formatting

- Naming conventions
  - Attributes, methods
  - Classes, packages
  - ...
- Design conventions
  - Collaborations
  - Modules, classes
  - ...
- Package structures
  - ...
- **See our Advanced Design and Programming (ADAP) course**

Quiz: Software Development

1. A file may have many authors. Should the names of these authors be listed in the file's header?
   - Yes
   - No

Build Process

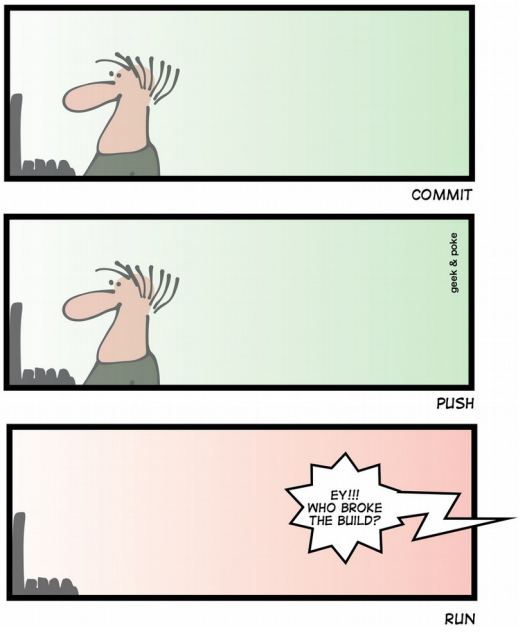
- Definition
  - Is the defined and (ideally automated) process of deriving an installable product from its source artifacts
- Purpose
  - Defines a standard environment
  - Provides developers with setup
  - Defines clear commit rules
  - Manages test data etc.

Developer Responsibilities

- Only work in defined standardized environment
  - This ensures no subtle differences to central build process
- Only commit / push code that compiles and works
  - “Breaking the build” through non-compiling code hurts team
- Only commit / push code that passes all the tests
  - Failing tests quickly degenerate system (hard to catch-up)

DEVELOPMENT CYCLE

FRIDAY EVENING EDITION



Build Asset Management

- Management by hand
  - Libraries are curated by hand
- Automated management
  - Libraries are pulled in automatically
- Bill of materials
  - Possibly generated automatically

Quiz: Supply Chain Effects

- Last week you delivered your product release to your client. The phone rings and an angry client is on the line, complaining about missing their schedule. What might have gone wrong?
  1. The client's development team objected
  2. The client's quality assurance unit objected
  3. The client's legal team objected
  4. All of the above

Review / Summary of Session

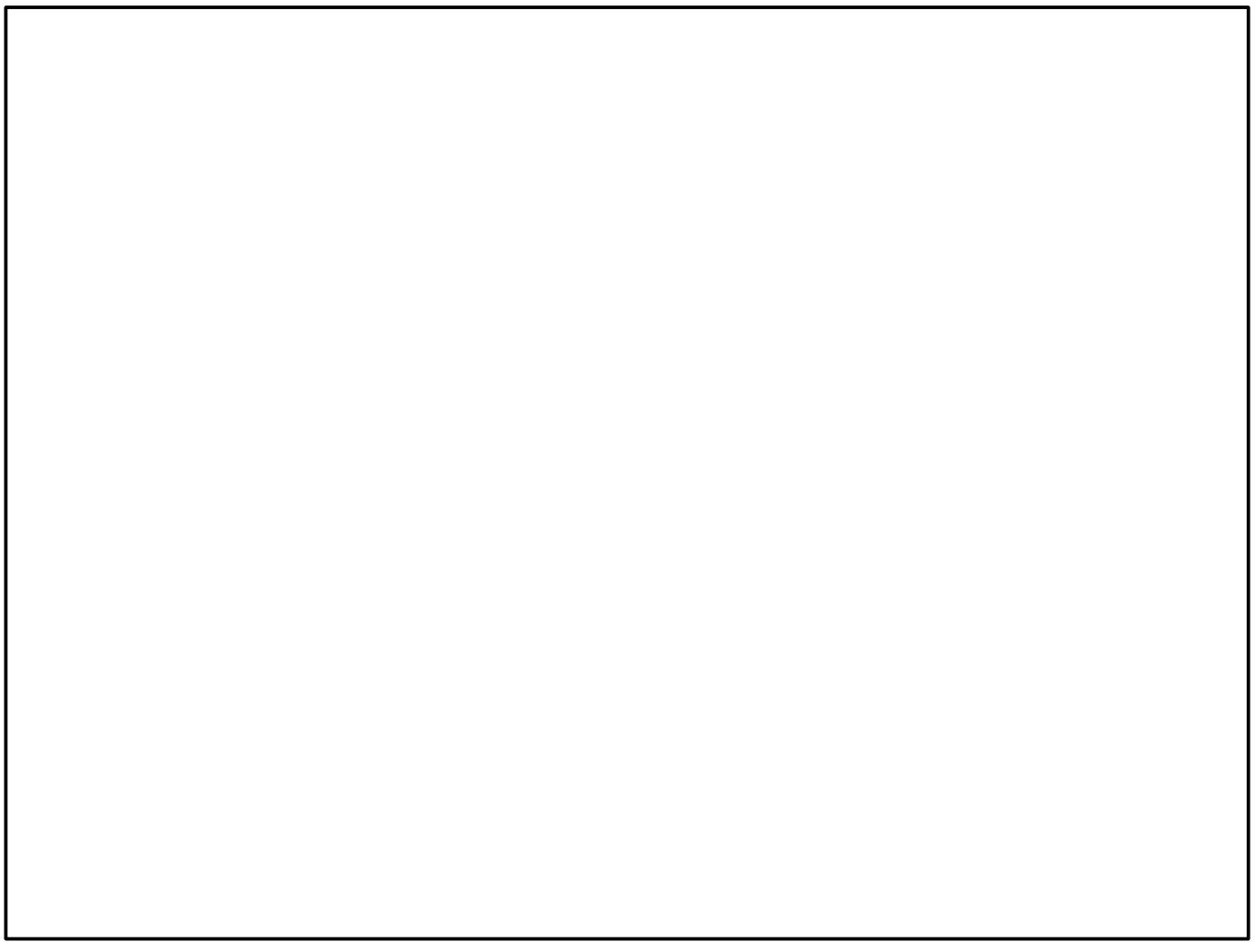
- Development planning
  - Stories vs. tasks
  - Daily scrum
- Source code and coding
  - Collective code ownership
  - Programming guidelines
  - Other conventions
- Build process

Thank you! Questions?

dirk.riehle@fau.de – http://osr.cs.fau.de

dirk@riehle.org – http://dirkriehle.com – @dirkriehle

DR



Credits and License

- Original version
  - © 2012-2019 Dirk Riehle, some rights reserved
  - Licensed under Creative Commons Attribution 4.0 International License
- Contributions
  - ...