

(Scrum) Software Development

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

AMOS E01

Licensed under [CC BY 4.0 International](#)

Agenda

1. Software developer
2. Sprint planning
3. Day planning
4. Code ownership
5. Build management

1. Software Developer

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?

Traditional to Scrum Role Mapping (Recap)

Traditional

Scrum

Product Manager

Product Owner

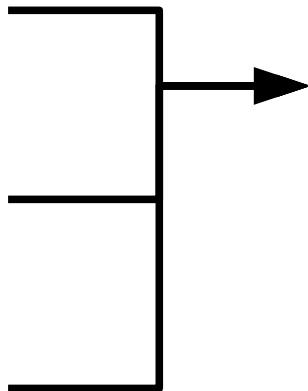
Engineering Manager

Software Developer

QA Engineer

Software Developer

Scrum Master



2. Sprint Planning

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

<div>14Product backlog+...</div> <div> <div> Show next photo... #107 opened by dirkriehle Est Size = 8 </div> <div> Rate photo and proceed... #106 opened by dirkriehle Est Size = 8 </div> <div> Show or hide photo caption... #108 opened by dirkriehle Est Size = 8 </div> <div> Create default admin... #110 opened by dirkriehle Est Size = 5 </div> <div> Startup and shutdown scripts... #122 opened by dirkriehle Est Size = 3 </div> <div> Reboot and shtudown UI... #123 opened by dirkriehle Est Size = 5 </div> <div> Cold backup and restore... #128 opened by dirkriehle Est Size = 8 </div> <div> Flag photo... #124 opened by dirkriehle Est Size = 8 </div> <div> View flagged photo queue... #125 opened by dirkriehle Est Size = 8 </div> <div> Review flagged photo... #126 opened by dirkriehle Est Size = 5 </div> <div> Track repeat offender... #127 opened by dirkriehle </div> </div>	<div>1Sprint backlog+...</div> <div> <div> Delete photo... #121 opened by dirkriehle Est Size = 5 </div> </div>	<div>2In progress+...</div> <div> <div> Select photo... #119 opened by dirkriehle Est Size = 5 </div> <div> Change photo data... #120 opened by dirkriehle Est Size = 3 </div> </div>	<div>1Awaiting review+...</div> <div> <div> Browse photo portfolio... #109 opened by dirkriehle Est Size = 8Real Size = 8 </div> </div>	<div>8Feature archive+...</div> <div> <div> Upload photo... #118 opened by dirkriehle Est Size = 8Real Size = 8 </div> <div> Change password... #117 opened by dirkriehle Est Size = 3Real Size = 5 </div> <div> Change basic profile... #116 opened by dirkriehle Est Size = 5Real Size = 5 </div> <div> Prompt basic profile... #115 opened by dirkriehle Est Size = 5Real Size = 5 </div> <div> Reset password... #114 opened by dirkriehle Est Size = 5Real Size = 5 </div> <div> Logout... #113 opened by dirkriehle Est Size = 3Real Size = 3 </div> <div> Login... #112 opened by dirkriehle Est Size = 5Real Size = 5 </div> <div> Register... #111 opened by dirkriehle Est Size = 8Real Size = 8 </div> </div>
--	--	---	--	--

▼ Actions

Board

Calendar

List

Gantt

status:open

▼ Filters

▼ Users

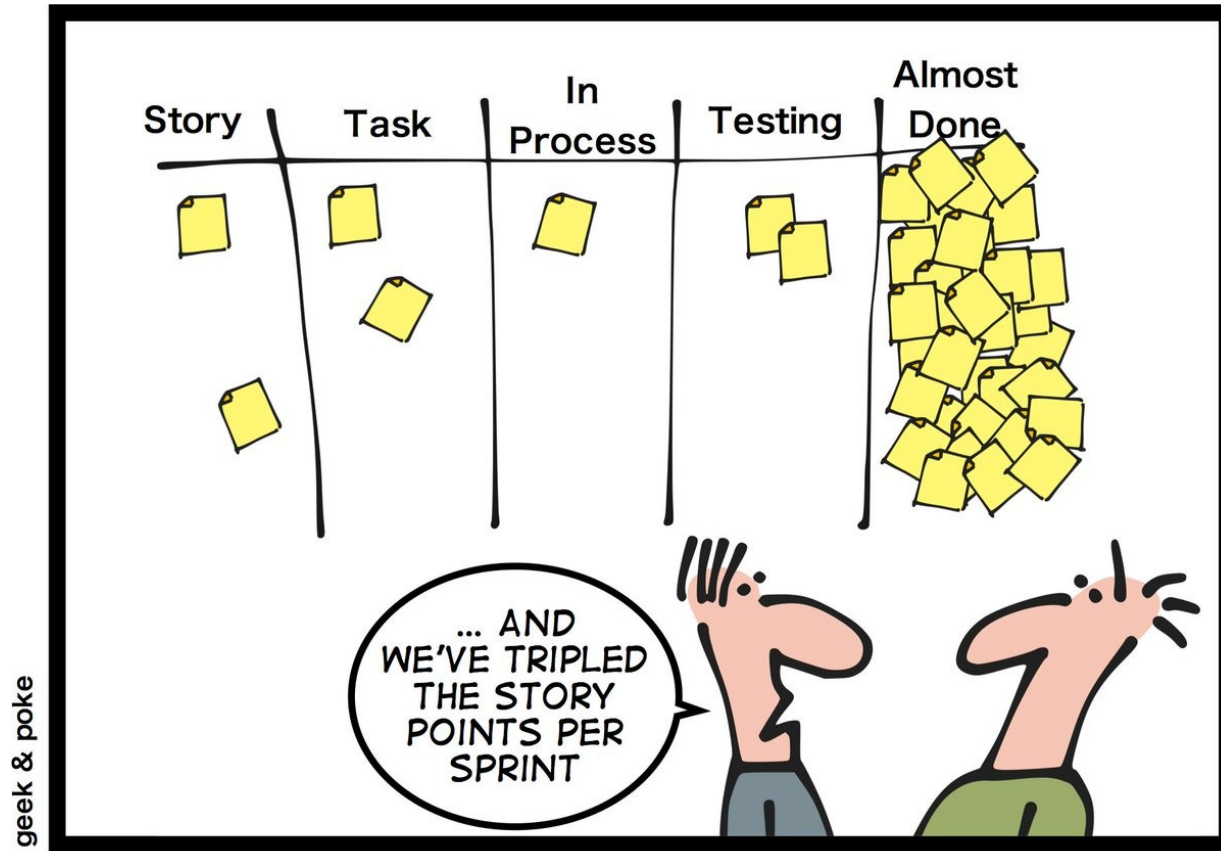
▼ Categories

Default swimlane	+ Backlog (6)	+ Ready (46)	+ Work in progress (0)	+ Done (0)
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)	<div>#133 Nobody assigned MultiProject: Get rid of deprecated methods and method calls</div> <div>#341 Nobody assigned Add feature to hide resources from certain commits</div> <div>#368 Nobody assigned HEAD pointer must be stored as foreign key to branch ref or commit id</div> <div>#370 Nobody assigned Think about NoSuchCommitException</div> <div>#403 Nobody assigned List<Project> getProjectsByProjectOwner(String projectOwnerName); should return list of names</div>	<div>#26 Nobody assigned Fully implement and test PageableIterator for users and projects</div> <div>#296 Hannes Dohrn Let WRI implement Comparable interface</div> <div>#388 Nobody assigned Implement list resources pagination when accessed via web api</div> <div>#391 Nobody assigned Add boolean switch to getProjects method that let's you decide whether you want to see deleted projects or not</div> <div>#185 Jonas Gröger Implement project specific roles</div> <div>#429 Nobody assigned</div>		<div>#151 Hannes Dohrn Draft design: Reimplement Sweble UI in REST+JS</div> <div>#176 Hannes Dohrn RFC: REST Server + Web Server</div> <div>#187 Hannes Dohrn UI design proposal</div>

Task Board (Artifact)

- Task board (a.k.a. Kanban board)
 - Visualizes the progress towards finishing features and the current sprint
 - Shows for each feature the progress of implementation
 - May break down work into hours

Definition of Almost Done



DOAD

The AMOS Project
© 2021 Dirk Riehle - Some Rights Reserved

3. Day Planning

Day Planning (Practices)

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



- 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



4. Code Ownership

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

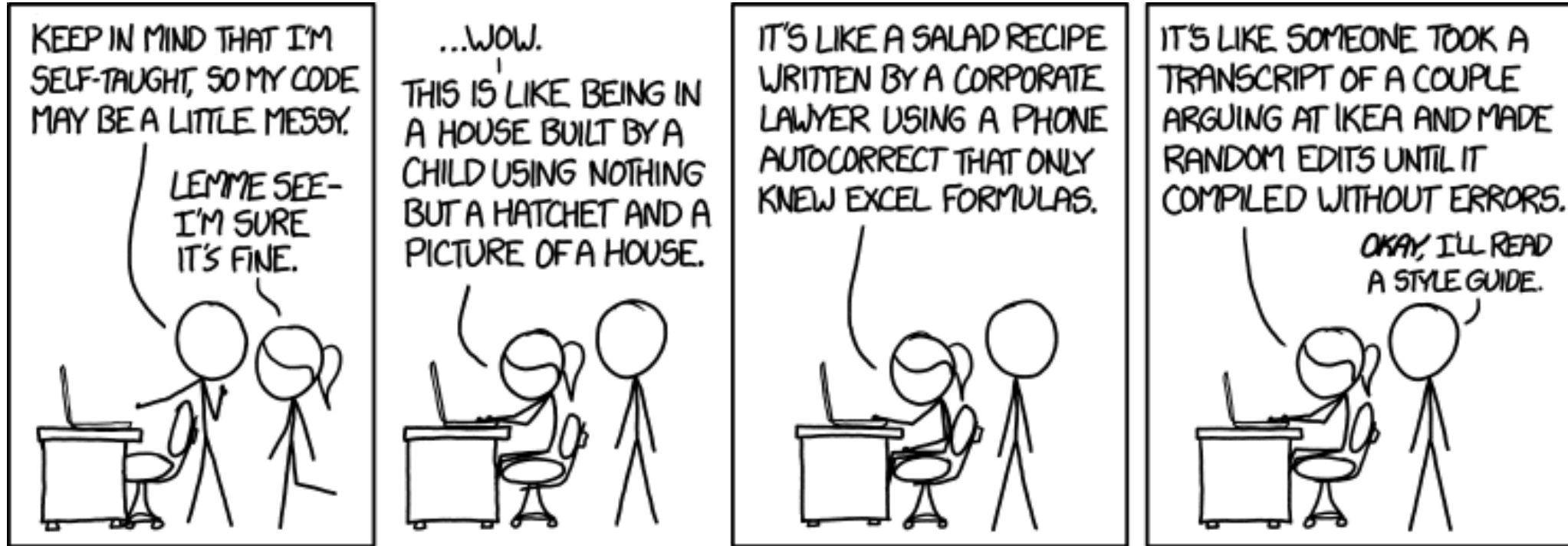
Programming Standard (Practice)

- 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
 */

...
```


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  
        ...  
    }  
}  
...
```

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	<code>void AccessControlContext#checkPermission(Permission)</code> throws <code>AccessControlException</code>
Name example	<code>void assertIsValidIndex(int)</code> throws <code>InvalidIndexException</code>
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

5. Build Management

Build Process

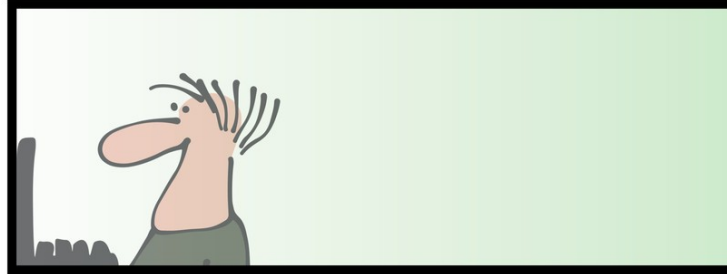
- Definition
 - Is the defined and (ideally automated) process of deriving an installable product from its source artifacts
- Purpose
 - Defines a standard environment
 - Provides developers with setup
 - Defines clear commit rules
 - Manages test data etc.

Developer Responsibilities

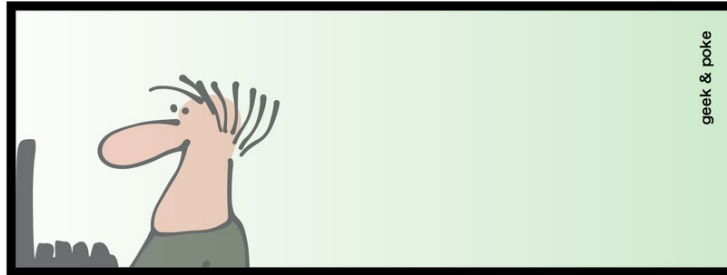
- Only work in defined standardized environment
 - This ensures no subtle differences to central build process
- Only commit / push code that compiles and works
 - “Breaking the build” through non-compiling code hurts team
- Only commit / push code that passes all the tests
 - Failing tests quickly degenerate 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
 - © 2021 Dirk Riehle, some rights reserved
 - Licensed under [Creative Commons Attribution 4.0 International License](#)
- Contributions
 - None yet