

TIC4302 - Information Security Practicum II

Agile (DevOps) Software Development with Open Source

The content taken from:

Software Engineering Practice Lecture by Prof. Dr. Dirk Riehle

https://oss.cs.fau.de/wp-content/uploads/2010/01/PSWT_WS200910_V20_Agile_Methods_Open_Source_DR_2010-01-20.pdf

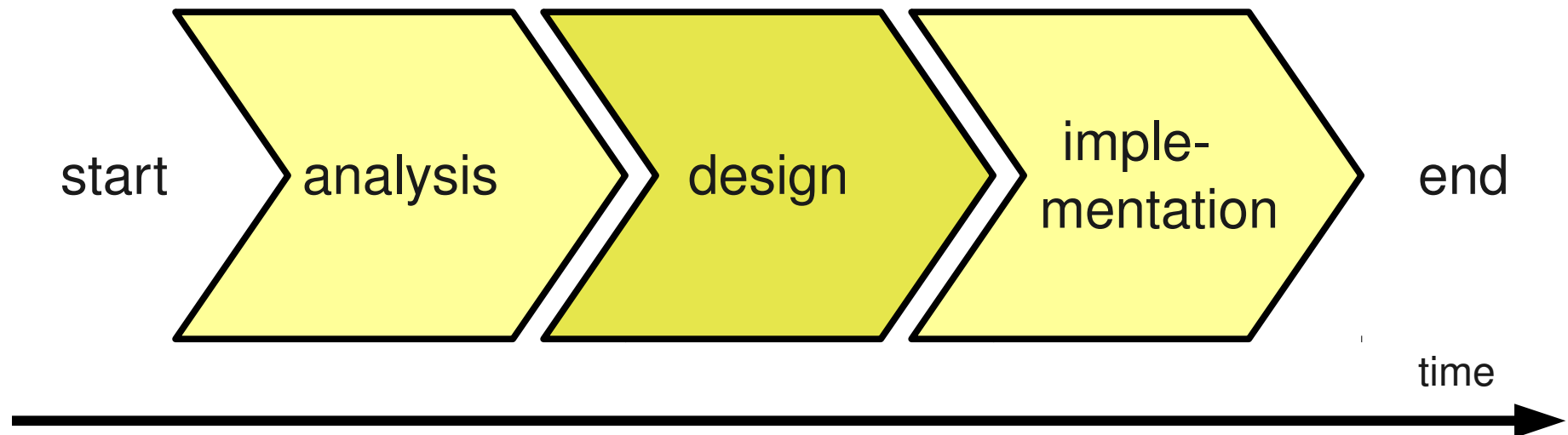
#	Content
1	Process Frameworks
2	Plan-Driven Software Development
3	Agile Methods (Scrum + Extreme Programming)
4	Open Source Software Development
5	Comparison of Frameworks
6	

Plan-Driven

Agile Methods

Open Source

- Linear, phase-oriented process
 - Main goal is to minimize risk through careful upfront planning
 - Equates phases with activities
 - No iterations, just one pass

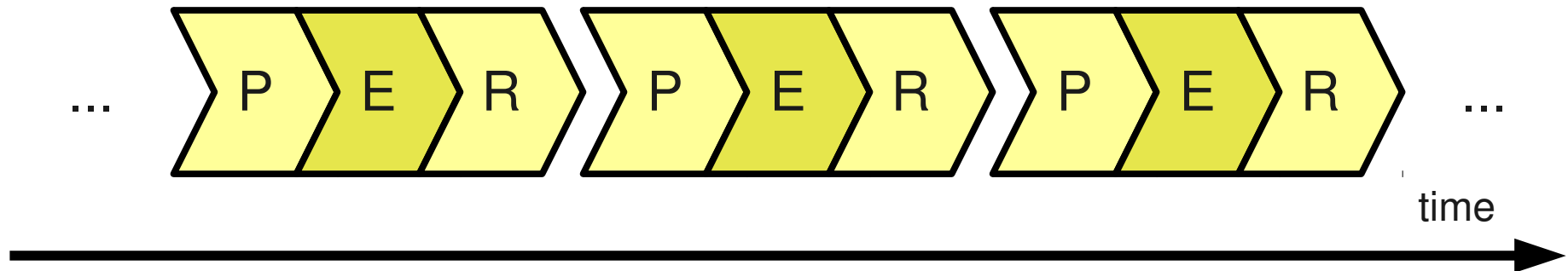


- Project negotiations
- Requirements analysis
- Contract definition
- System analysis
- Software architecture design
- System design
- User interface design
- Implementation
- System test
- Acceptance test
- Handover
- Actual use
- Lawsuit



**no feedback
until after
delivery**

- Agile methods is the name of a class of process frameworks
 - Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, etc.
 - Unified by the recognition of a common philosophical base and joined in their rejection of the traditional life-cycle model



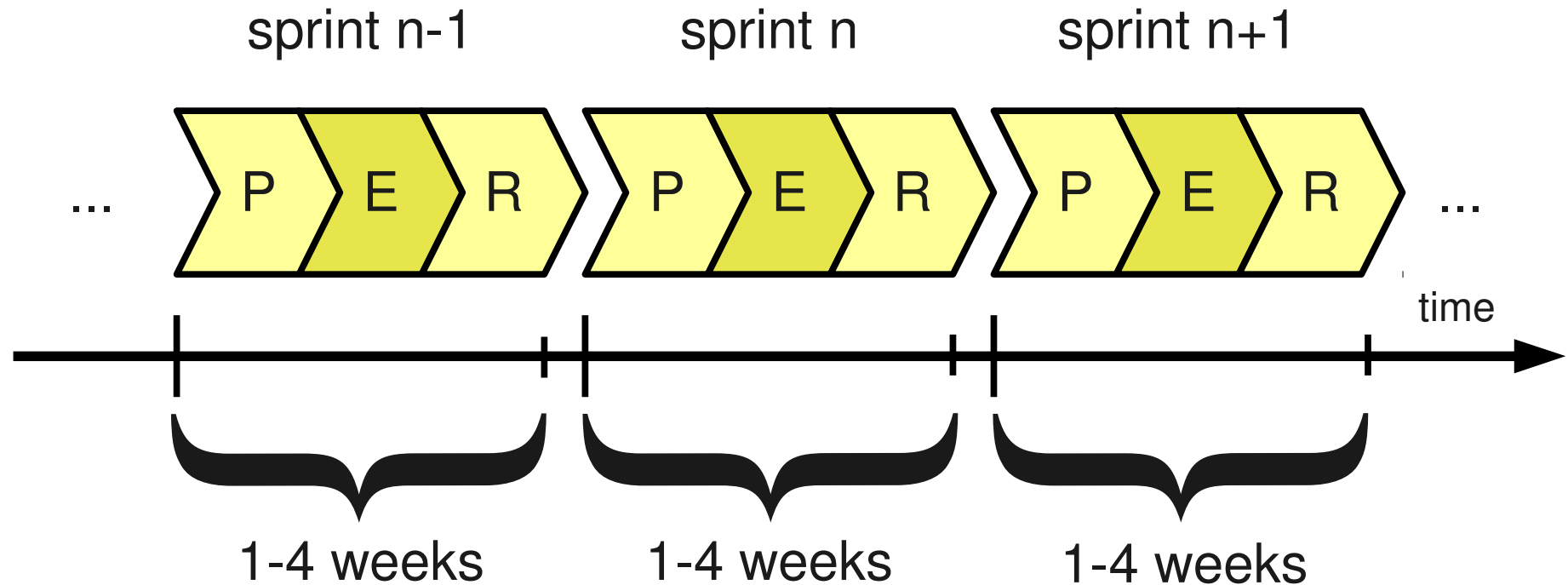
Scrum

(mostly as process framework)

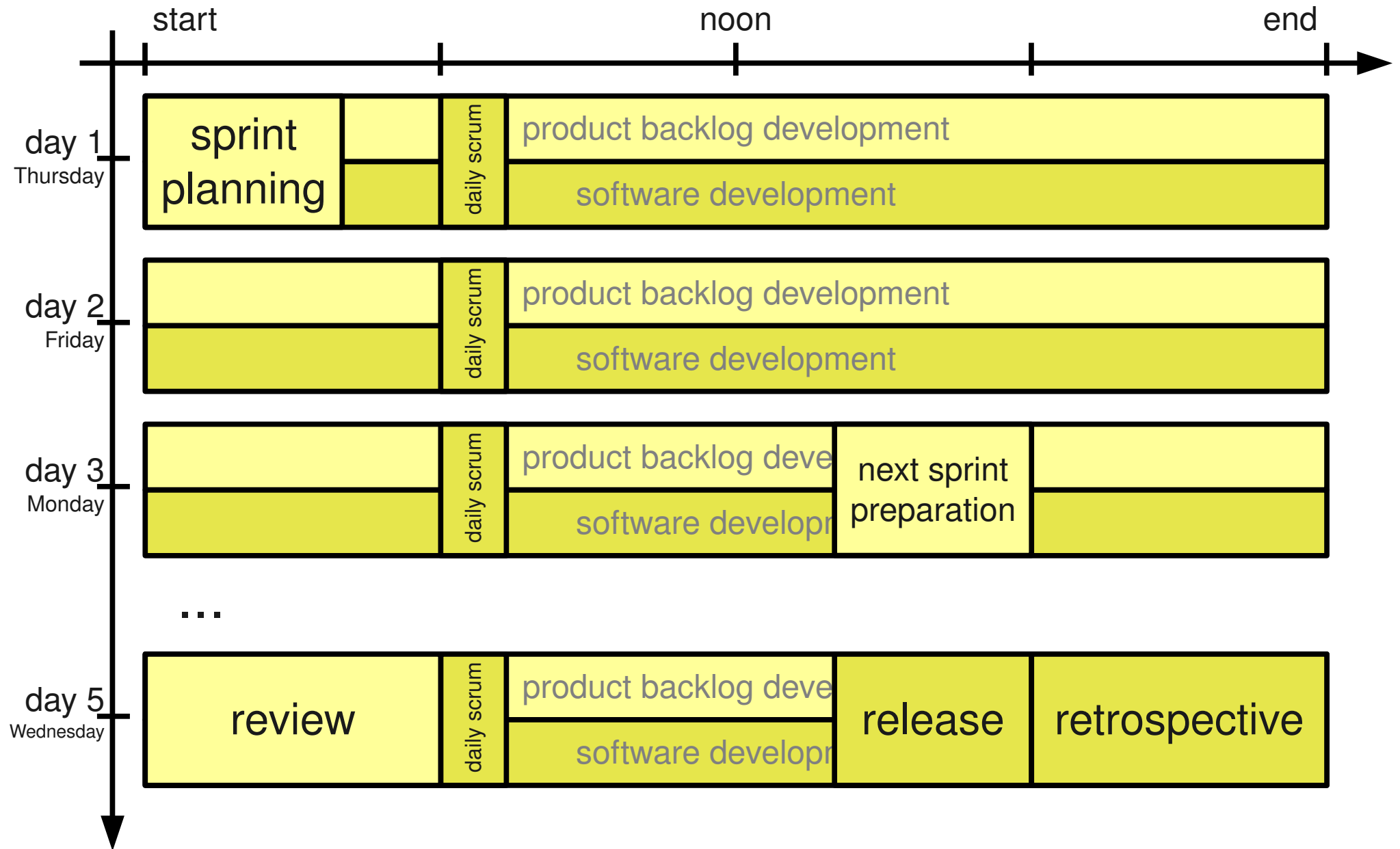
Extreme Programming (XP)

(mostly for development practices)

- Succession of **equal-length** sprints (= short iterations)
- Intervention points are during planning and review
- Product owner is always available to answer questions

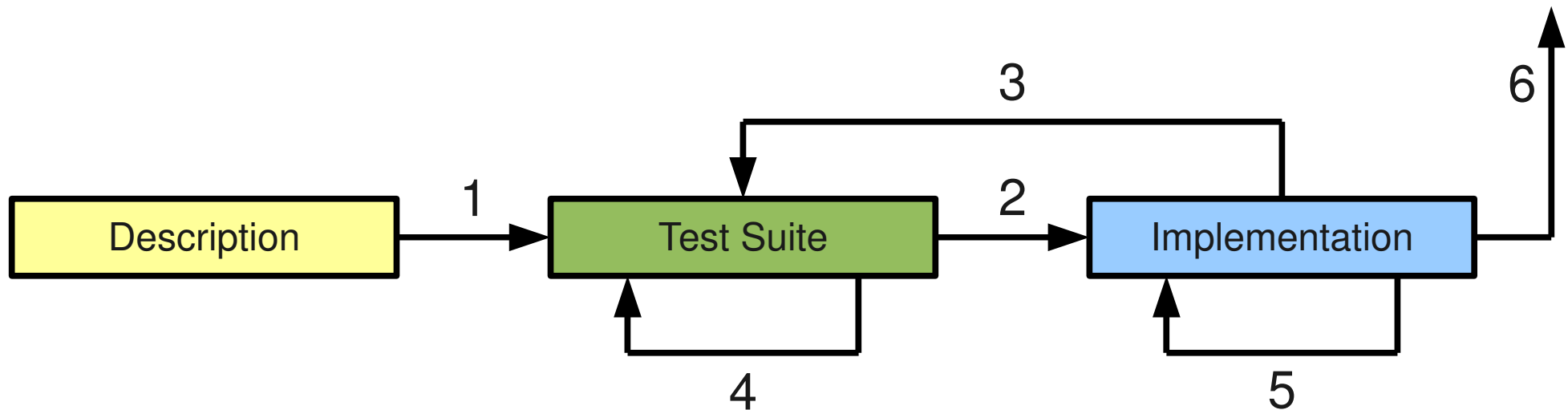


P = planning - E = engineering - R = review (or release or retrospective)



- Test-driven development (abbrev. TDD)
 - Is a core process for programming activities
 - Promises minimal fluctuations in development speed
 - Takes the stress out of refactoring your code base
 - Considered a high-end practice (requires too much discipline for many)
 - Requires significant experience with ancillary tools and techniques
- First described by Kent Beck in [Beck 2003]

1. Translate description into test suite
2. Implement feature to fulfill (“green-bar”) test suite
3. Revise test suite from new domain and implementation insights
4. Refactor test suite to keep code healthy
5. Refactor implementation to keep design and code healthy
6. Exit when test suite is complete and all tests succeed



Two simple rules

only write new code when a test fails

eliminate waste

deliver

clean code that works

Definition of Open Source (Abbreviated)

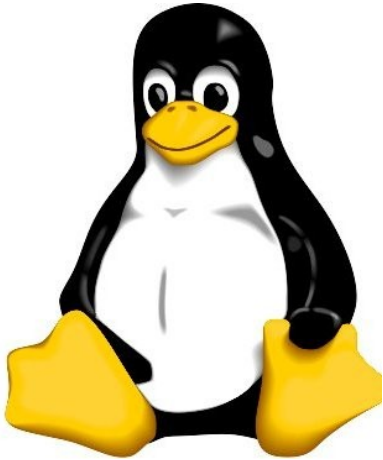
8.1

Open Source
Definition and History

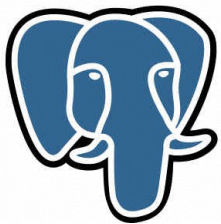
- Open source software is software that is
 - available in source code form
 - can be freely used and modified
 - can be freely redistributed
 - cf. four freedoms of software
- The open source initiative
 - maintains the definition and trademark
 - approves open source licenses



Some Open Source Projects

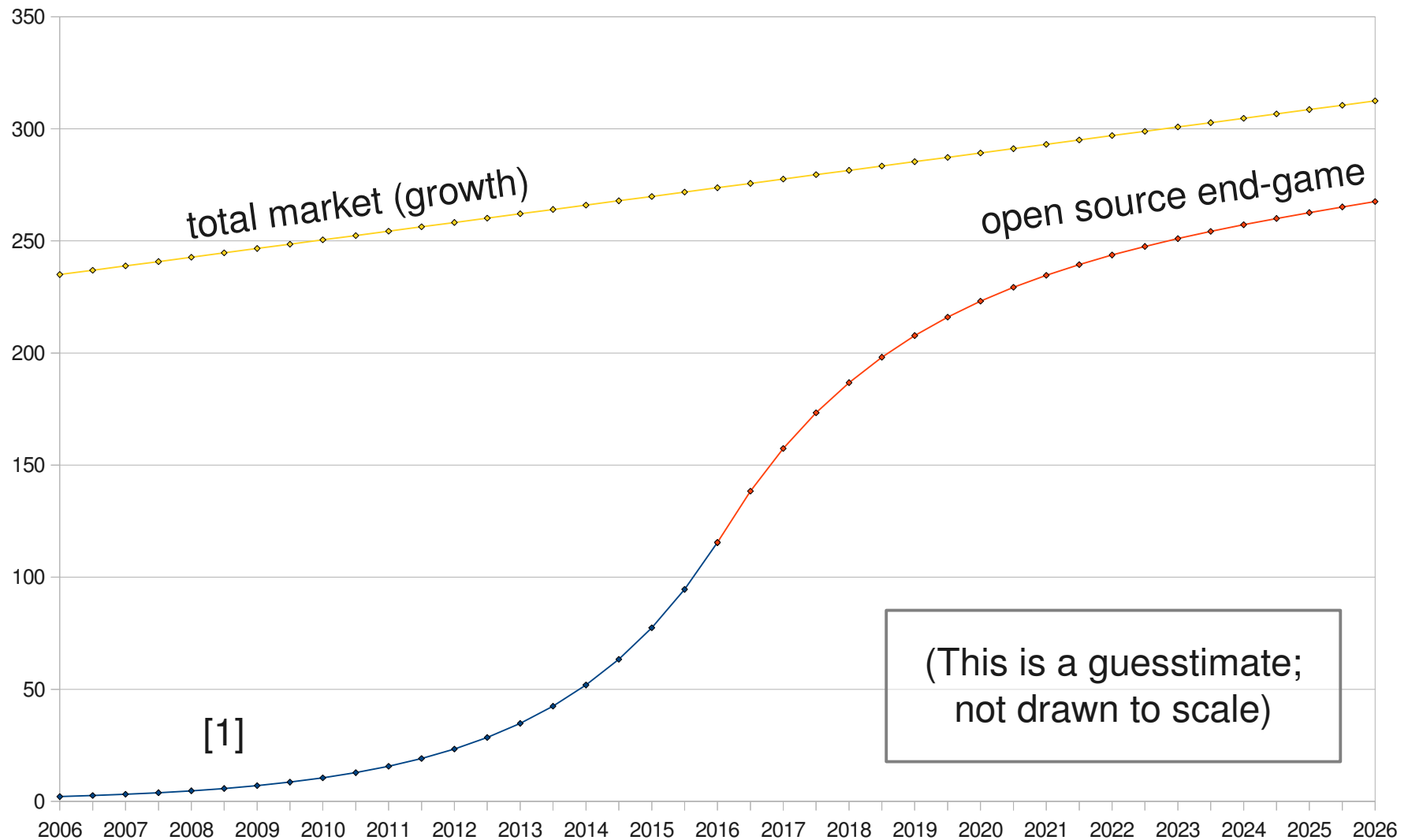


PostgreSQL



- The free software movement
 - Initiated (and still led) by Richard Stallman (MIT) in the 1980 to free software (from being closed)
 - Free software philosophy summarized as “free as in 'free speech' not as in 'free beer'”
- Free Software Foundation
 - <http://www.fsf.org>
 - U.S. 501(c) non-profit organization
- The open source movement
 - Eric Raymond: “Cathedral and the Bazaar” --- describes open source as development method
 - Formalized 1998 to address perceived anti-commerce bias of “free software”
- Open Source Initiative
 - <http://www.opensource.org>
 - U.S. 501(c) non-profit organization

FLOSS = Free/Libre/Open Source Software



[1] Deshpande, Riehle. "The Total Growth of Open Source." In Proceedings of OSS 2008.

	Community-owned	Single or dominant proprietor
Single product or product line	Community Open Source (e.g. Linux, TikiWiki)	Commercial Open Source (e.g. MySQL, Jasper)
Multi-product assembly ("stack")	Community Distribution (e.g. Debian)	Commercial Distribution (e.g. RHEL, SLES)

1991: Linux project started

1998: Open Source Initiative founded

Traditional Community Open Source

1999: Apache Software Foundation founded

2004: Eclipse Foundation founded

Managed Community Open Source

1995: MySQL AB founded

2001: MySQL AB funded

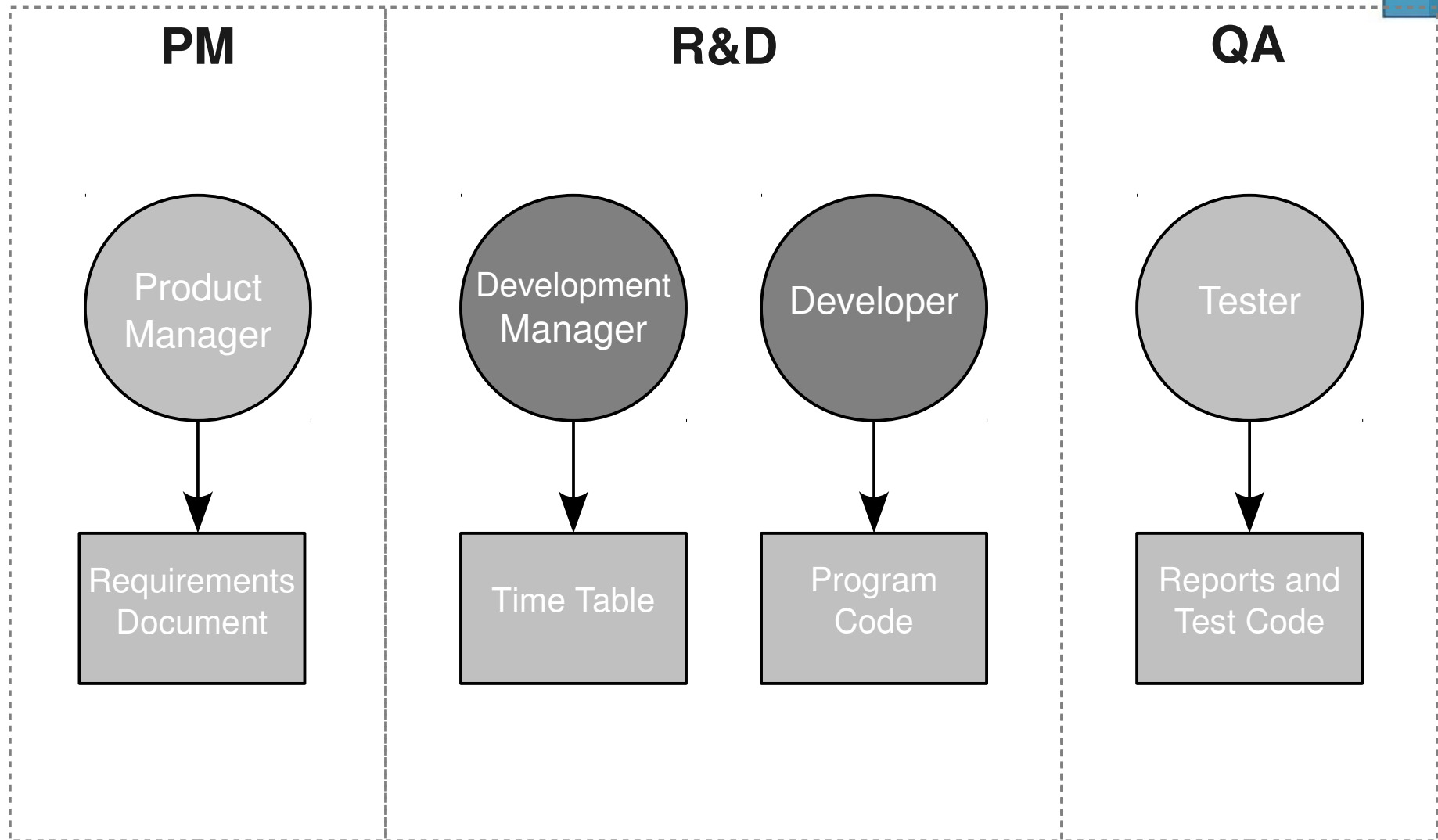
Single Vendor ("Commercial") Open Source

Traditional Work

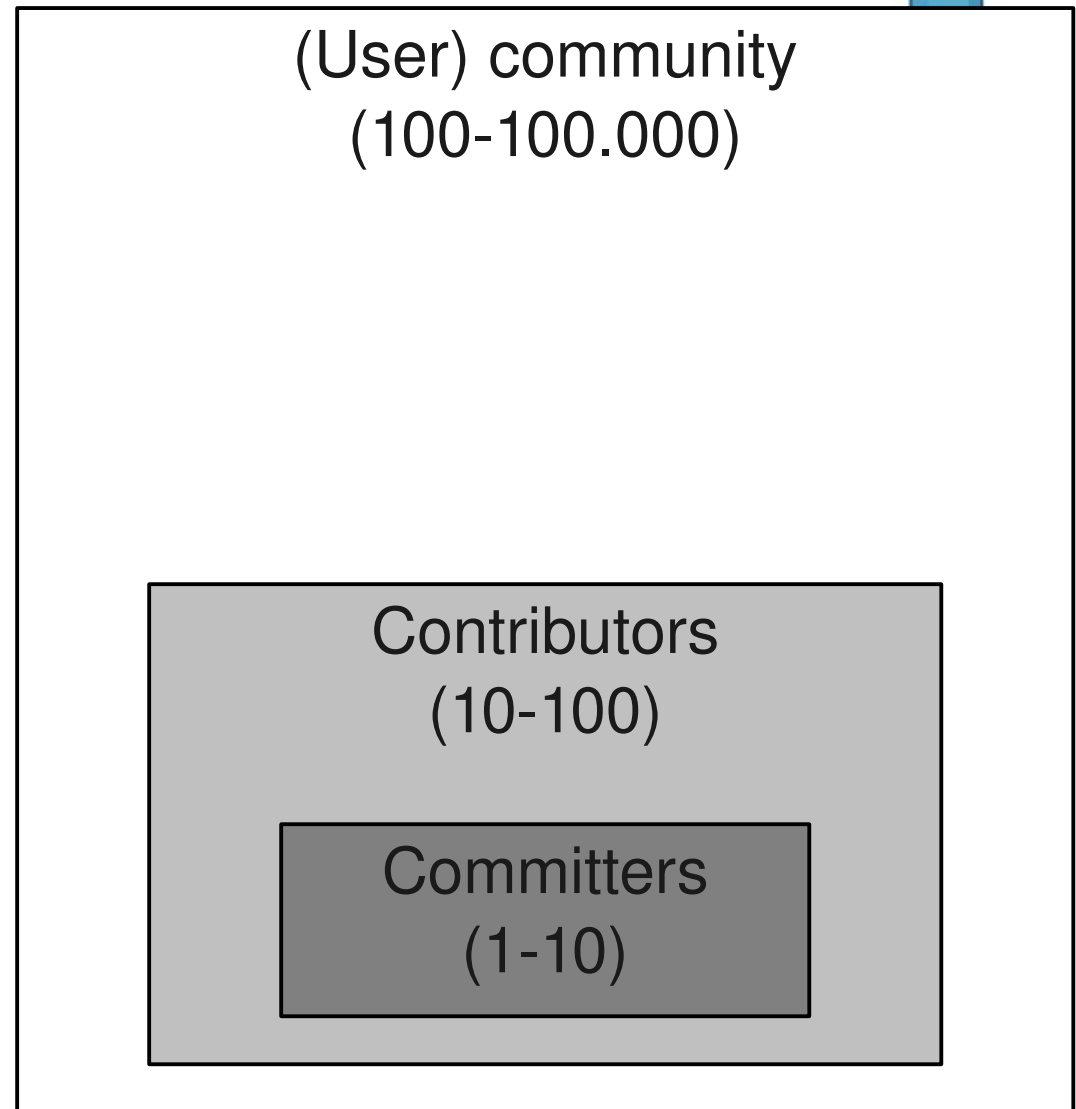
- Hierarchical
 - Closed and hidden silos
 - Assigned to project
- Status-driven
 - Public and private discussions
 - Hierarchical status decides
- Assigned tasks
 - Prescribed process
 - Prescribed jobs

Open Collaboration

- Egalitarian
 - Open for contribution
 - Everyone can contribute
- Meritocratic
 - Public discussion process
 - Decisions based on merit
- Self-organizing
 - People find their own process
 - People find their best project



- Committers
 - Main developers
 - With access rights
- Contributors
 - Casual developers
 - Submit patches
- (User) community
 - Provide bug reports
 - Provide feedback



		new application domain	
		no	yes
large project	no	plan-driven agile methods open source	agile methods open source
	yes	plan-driven open source	agile methods open source



Questions? Feedback!

<http://dirkriehle.com> - dirk.riehle@cs.fau.de - [@dirkriehle](https://twitter.com/dirkriehle)