
# Lappy 386

# #chaq16

Universiteit Antwerpen

AGENTSCHAP INNOVEREN & ONDERNEMEN

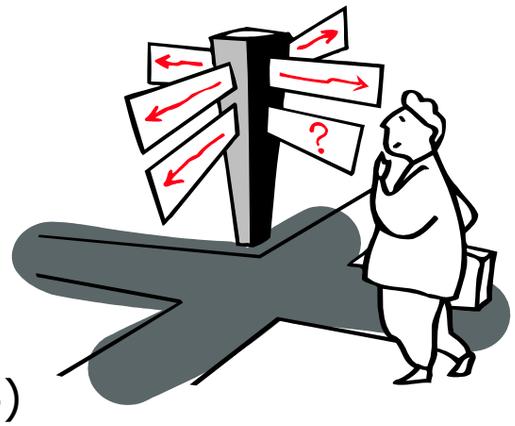
Vrije Universiteit Brussel

# CHA-Q

## Change-centric quality Assurance

December 2016

# Agenda



[12:30 - 13:00] — Registration

[13:00 - 13:30] — Welcome Prof. S. Demeyer (UA) and Prof. C. De Roover (VUB)
[13:30 - 14:20] — Pilot Studies on Cha-Q Technology for "Mining and Automating Past Changes"
- Mining Git Repositories for Repeated Systematic Edits at TPVision - Reinout Stevens (VUB)
- Automating Systematic Edits at FOD Financiën - Tim Molderez (VUB)

[14:20 - 14:30] — Coffee break

[14:30 - 14:55] — Pilot Studies on Cha-Q Technology for "Monitoring Ongoing Changes"
- Maintaining Explicit Inter-Artefact Links at Inventive Designers - Angela Lozano (VUB)

[14:55 - 15:45] — Pilot Studies on Cha-Q Technology for "History-Inspired Decision Making"
- Exploiting Issue Tracker for Project Planning at Inventive Designers - Murgia Alessandro (UA)
- Strengthening the Regression Test Suite at HealthConnect - Ali Parsai - (UA)

[15:45 - 15:55] — Coffee break

[15:55 - 16:45] — Invited Talk
"What Software Analytics Can Do for Developers and Testers"
Andy Zaidman (Delft University of Technology, The Netherlands)

[16:45 - 18:00] — Poster Reception and Networking Drink

# Practical Arrangements

# Practical Arrangements

# Practical Arrangements


# Program Issues

# Program Issues


# Program Issues


# Technology Readiness Level

- **TRL 0** — *Idea.* Unproven concept, no testing has been performed
- **TRL 1** — *Basic research.* Principles postulated and observed; no experimental proof available
- **TRL 2** — *Technology formulation.* Concept and application have been formulated
- **TRL 3** — *Applied research.* First laboratory tests completed; proof of concept
- **TRL 4** — *Small scale prototype* built in a laboratory environment ("ugly" prototype)
- **TRL 5** — *Large scale prototype tested* in intended environment
- **TRL 6** — *Prototype system tested* in intended environment close to expected performance
- **TRL 7** — *Demonstration system* operating in operational environment at pre-commercial scale
- **TRL 8** — *First of a kind commercial* system. Manufacturing issues solved
- **TRL 9** — *Full commercial application*, technology available for consumers.

# Technology Readiness Level

AGENTSCHAP INNOVEREN & ONDERNEMEN

- **TRL 0** — *Idea.* Unproven concept, no testing has been performed
- **TRL 1** — *Basic research.* Principles postulated and observed; no experimental proof available
- **TRL 2** — *Technology formulation.* Concept and application have been formulated
- **TRL 3** — *Applied research.* First laboratory tests completed; proof of concept
- **TRL 4** — *Small scale prototype* built in a laboratory environment ("ugly" prototype)
- **TRL 5** — *Large scale prototype tested* in intended environment
- **TRL 6** — *Prototype system tested* in intended environment close to expected performance
- **TRL 7** — *Demonstration system* operating in operational environment at pre-commercial scale
- **TRL 8** — *First of a kind commercial* system. Manufacturing issues solved
- **TRL 9** — *Full commercial application*, technology available for consumers.

# Partners

AGENTSCHAP INNOVEREN & ONDERNEMEN

Universiteit Antwerpen

VRIJE UNIVERSITEIT BRUSSEL

BARCO Visibly yours

ctg

TPVISION

MediaGeniX

jamo solutions

inventivedesigners the output innovators

Federale Overheidsdienst FINANCIEN

health CONNECT ICT EXPERTISE THAT CARES

OMPARTNERS

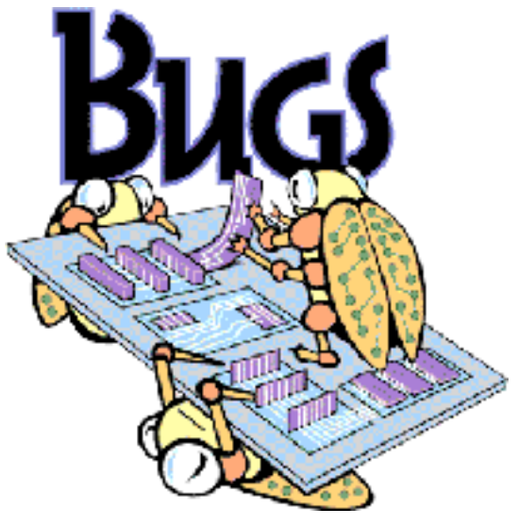
# WHY ?


Changes
Past
annotations
recurring
program
misclassified
Analyzing
Present
Software
change-aware
reports
mining
Repeating
bug
links
re
long
transformations
traceability
meta
Change-Centric
Future
logging
meta-data
model
active
fix
evolution
API
process
distilling
monitor
bug-fixes
Development
Tracing
test

# Reliability vs. Agility

Software is vital to our society ⇒ Software must be reliable

Traditional Software Engineering
Reliable = Software without bugs

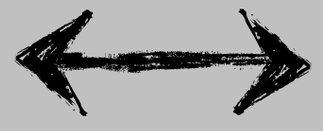



Today's Software Engineering
Reliable = Easy to Adapt


*On the Origin of Species*

Striving for RELIABILITY

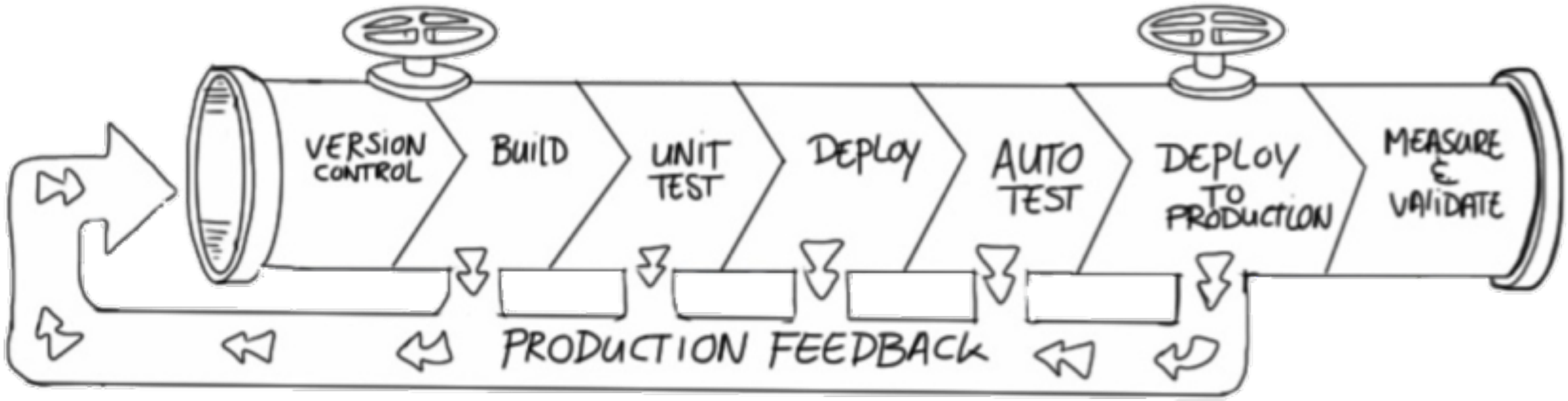
(Optimise for *perfection*)



Striving for AGILITY

(Optimise for *development speed*)

# Continuous Integration / Deployment



# Software Repositories & Archives

**Version Control**

- CVS, Subversion, Git, …
- Rational ClearCase
- Perforce,
- Visual Source Safe
- …

**Automate the Build**

- make
- Ant, Maven
- MSBuild
- OpenMake
- Build Forge
- …

**Issue Tracking**

- Bugzilla
- BugTracker.NET
- ClearQuest
- JIRA
- Mant
- Visual Studio Team Foundation Server
- …

**Automated Testing**

- HP QuickTest Professional
- IBM Rational Functional Tester
- Maveryx
- Selenium
- TestComplete
- Visual Studio Test Professional Microsoft 2010
- …

**… mailing archives, newsgroups, chat-boxes, facebook, twitter, …**

# Software Repositories & Archives

**Version Control**

- CVS, Subversion, Git, …
- Rational ClearCase
- Perforce,
- Visual Source Safe
- …

**Automate the Bui**

- make
- Ant, Maven
- MSBuild
- OpenMake
- Build Forge

*All of a sudden empirical research has what any empirical science needs: a large corpus of objects to analyze.*

[Bertrand Meyer's technology blog]

**Issue Tracking**

- Bugzilla
- BugTracker.NET
- ClearQuest
- JIRA
- Mant
- Visual Studio Team Foundation Server
- …

**Automated Testing**

- HP QuickTest Professional
- IBM Rational Functional Tester
- Maveryx
- Selenium
- TestComplete
- Visual Studio Test Professional Microsoft 2010
- …

**… mailing archives, newsgroups, chat-boxes, facebook, twitter, …**

# Mining Software Repositories

*The Mining Software Repositories (MSR) field analyzes the rich data available in software repositories to uncover interesting and actionable information about software systems and projects.*

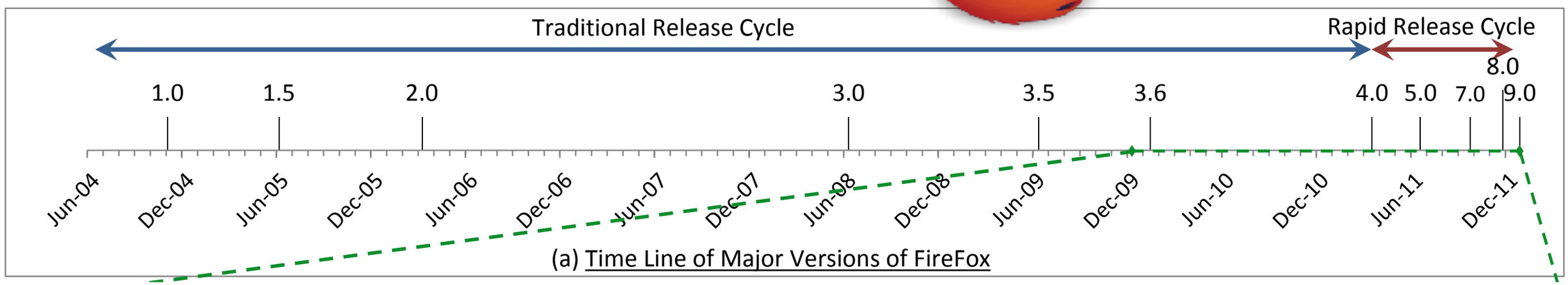
**Conferences**

- 2017—14th edition, Buenos Aires, Argentina
- 2016—13th edition, Austin, Texas
- 2015—12th edition, Florence, Italy
- 2014—11th edition, Hyderabad, India
- 2013—10th edition, San Francisco, CA, USA
- 2012—9th edition, Zürich, CH
- 2011—8th edition, Honolulu, HI, USA
- 2010—7th edition, Cape Town, ZAF
- 2009—6th edition, Vancouver, CAN
- 2008—5th edition, Leipzig, DEU
- 2007—4th edition, Minneapolis, MN, USA
- 2006—3rd edition, Shanghai, CHN
- 2005—2nd edition, Saint Luis, MO, USA
- 2004—1st edition, Edinburgh, UK

**Hall of Fame—Mining Challenge**

- 2017 — TravisTorrent (Github)
- 2016 — BOA (SourceForge & Github)
- 2015 — StackOverflow
- 2014—Sentiment Analysis of Commit Messages in *GitHub*: An Empirical Study
- 2013—Encouraging User Behaviour with Achievements: An Empirical Study [*StackOverflow*]
- 2012—Do the Stars Align? Multidimensional Analysis of *Android's* Layered Architecture
- 2011—Apples Vs. Oranges? An exploration of the challenges of comparing the source code of two software systems [Netbeans+Eclipse]
- 2010—Cloning and Copying between *GNOME* Projects
- 2009—On the use of Internet Relay Chat (IRC) meeting by developers of the *GNOME* GTK+ project
- 2008—A newbie's guide to *Eclipse* APIs
- 2007—Mining *Eclipse* Developer Contributions via Author-Topic Models
- 2006—A study of the contributors of *PostgreSQL*

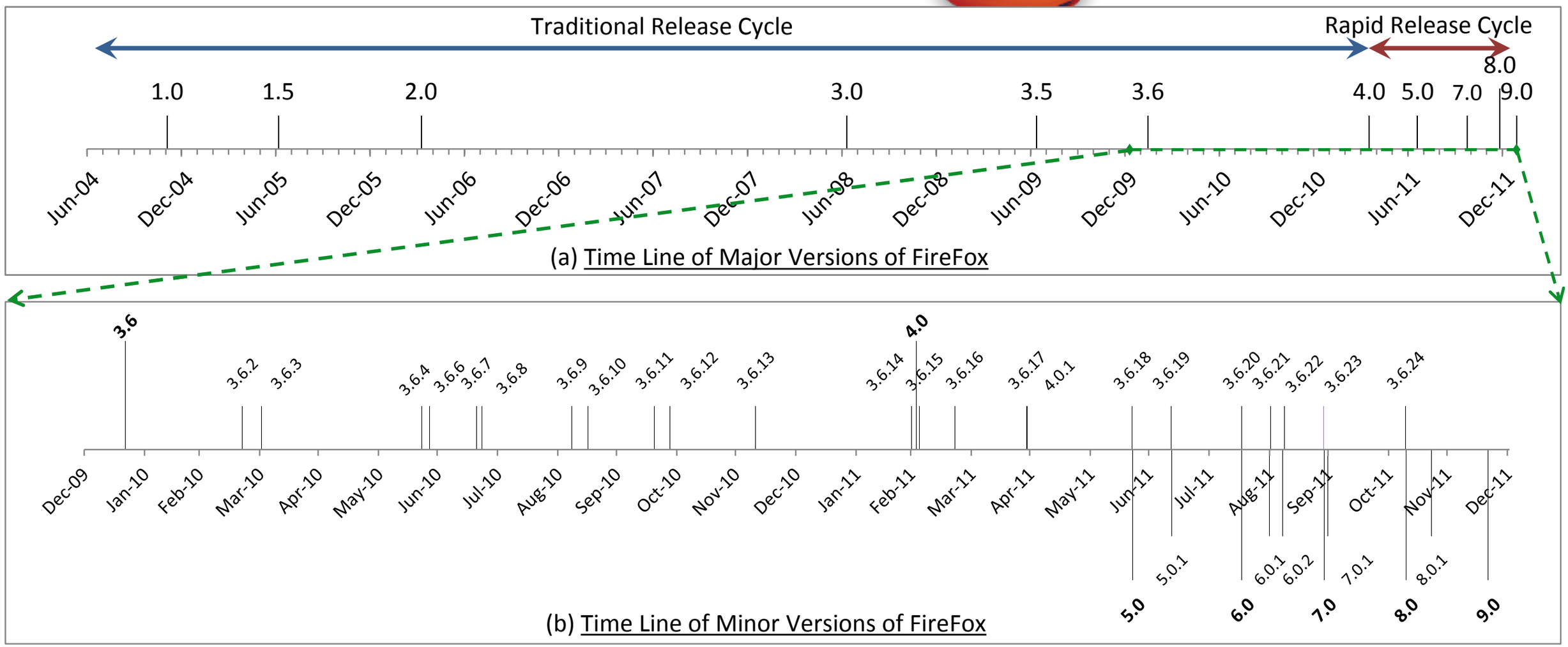
# The Case of Firefox (1/2)




(a) Time Line of Major Versions of FireFox

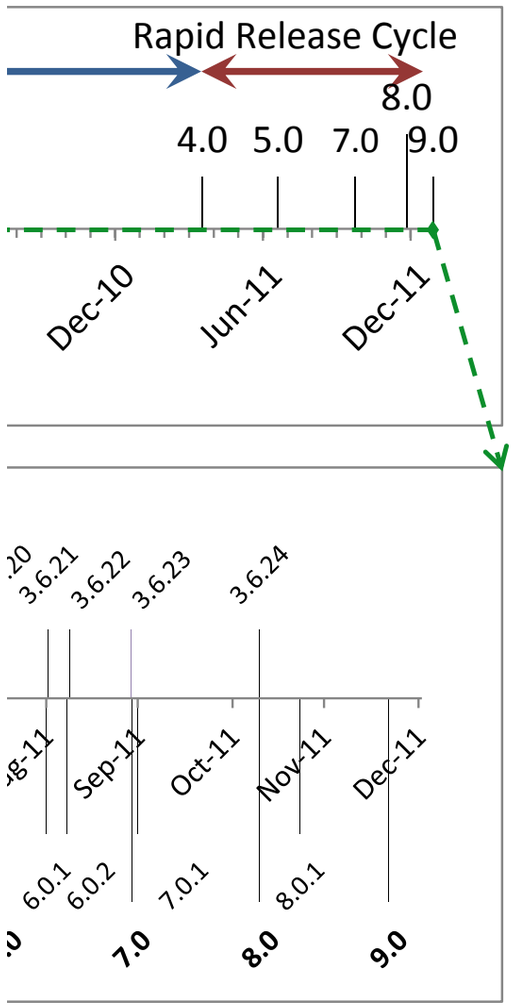
[Khom2014] Khomh, F. Adams, B, Dhaliwal, T and Zou, Y Understanding the Impact of Rapid Releases on Software Quality: The Case of Firefox, Empirical Software Engineering, Springer. http://link.springer.com/article/10.1007/s10664-014-9308-x

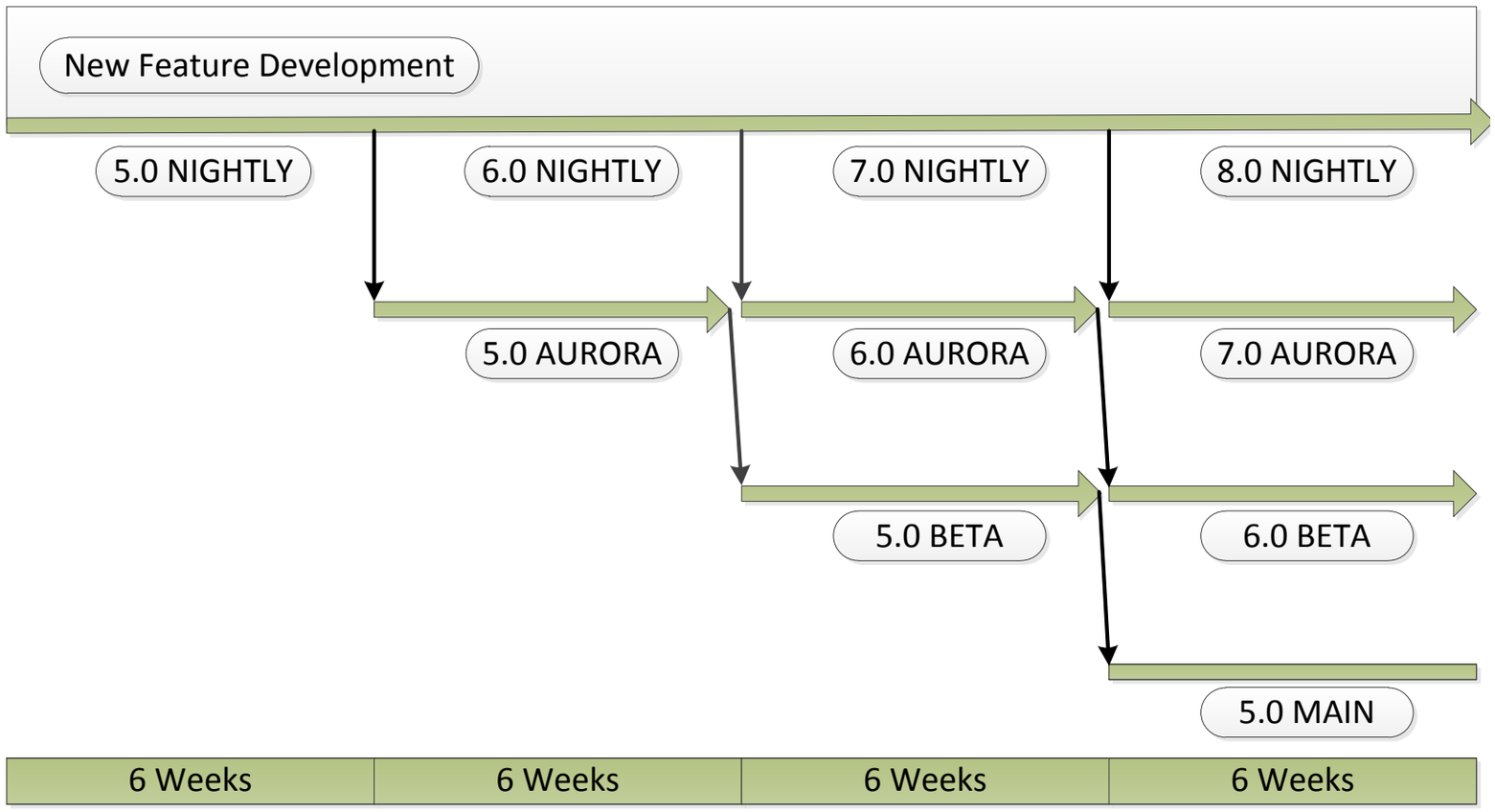
# The Case of Firefox (1/2)




[Khom2014] Khomh, F. Adams, B, Dhaliwal, T and Zou, Y Understanding the Impact of Rapid Releases on Software Quality: The Case of Firefox, Empirical Software Engineering, Springer.
http://link.springer.com/article/10.1007/s10664-014-9308-x

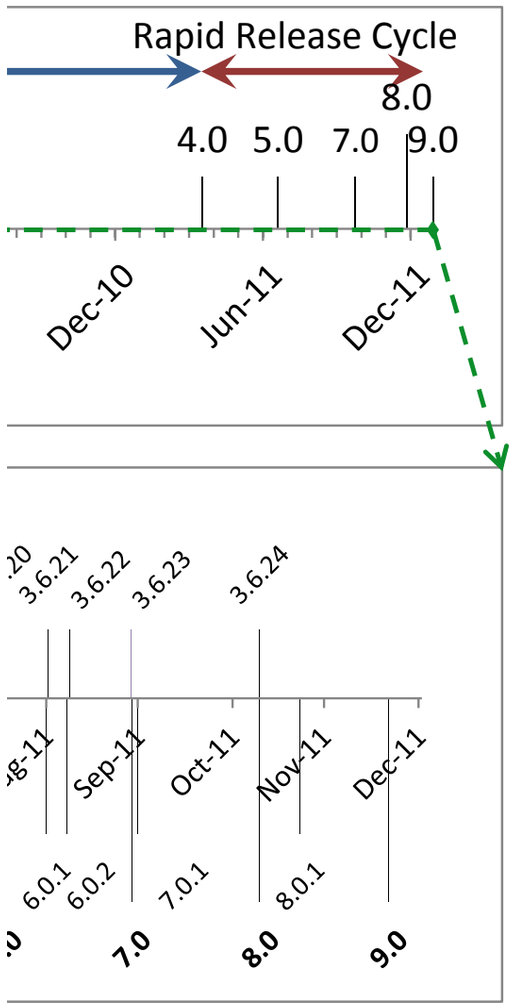
# The Case of Firefox (2/2)

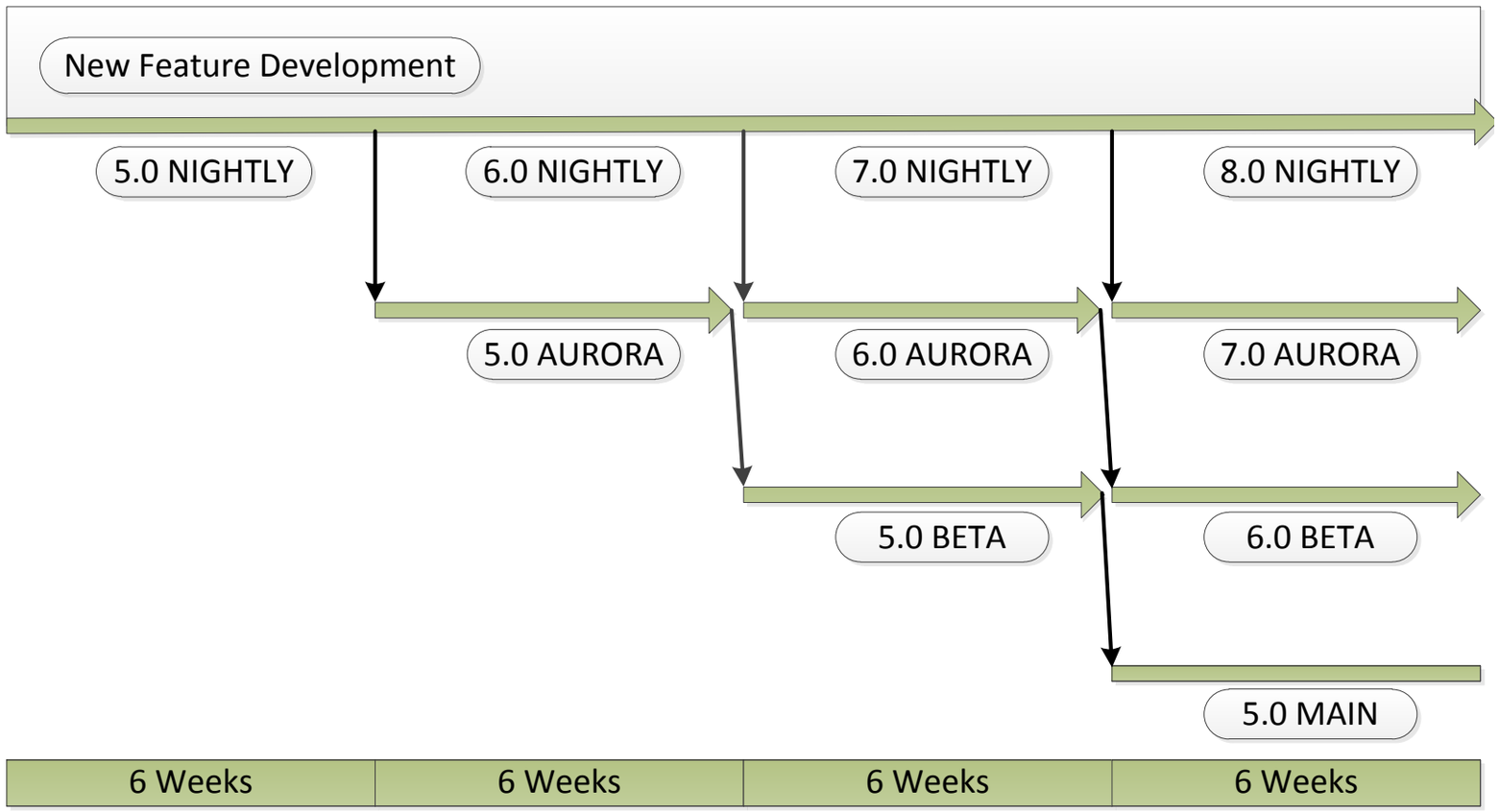



[Khom2014] Khomh, F. Adams, B, Dhaliwal, T and Zou, Y Understanding the Impact of Rapid Releases on Software Quality: The Case of Firefox, Empirical Software Engineering, Springer. http://link.springer.com/article/10.1007/s10664-014-9308-x

# The Case of Firefox (2/2)



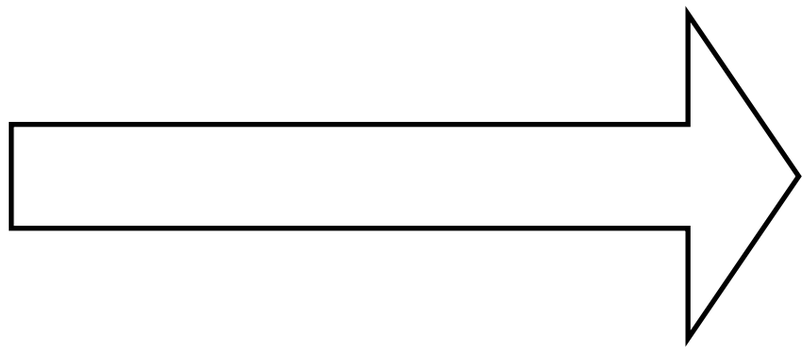




- ✓ bugs are fixed faster
  (but … harder bugs propagated to later releases)
- ✓ amount of pre- & post-release bugs ± the same
- ✓ the program crashes earlier
  (perhaps due to recent features)



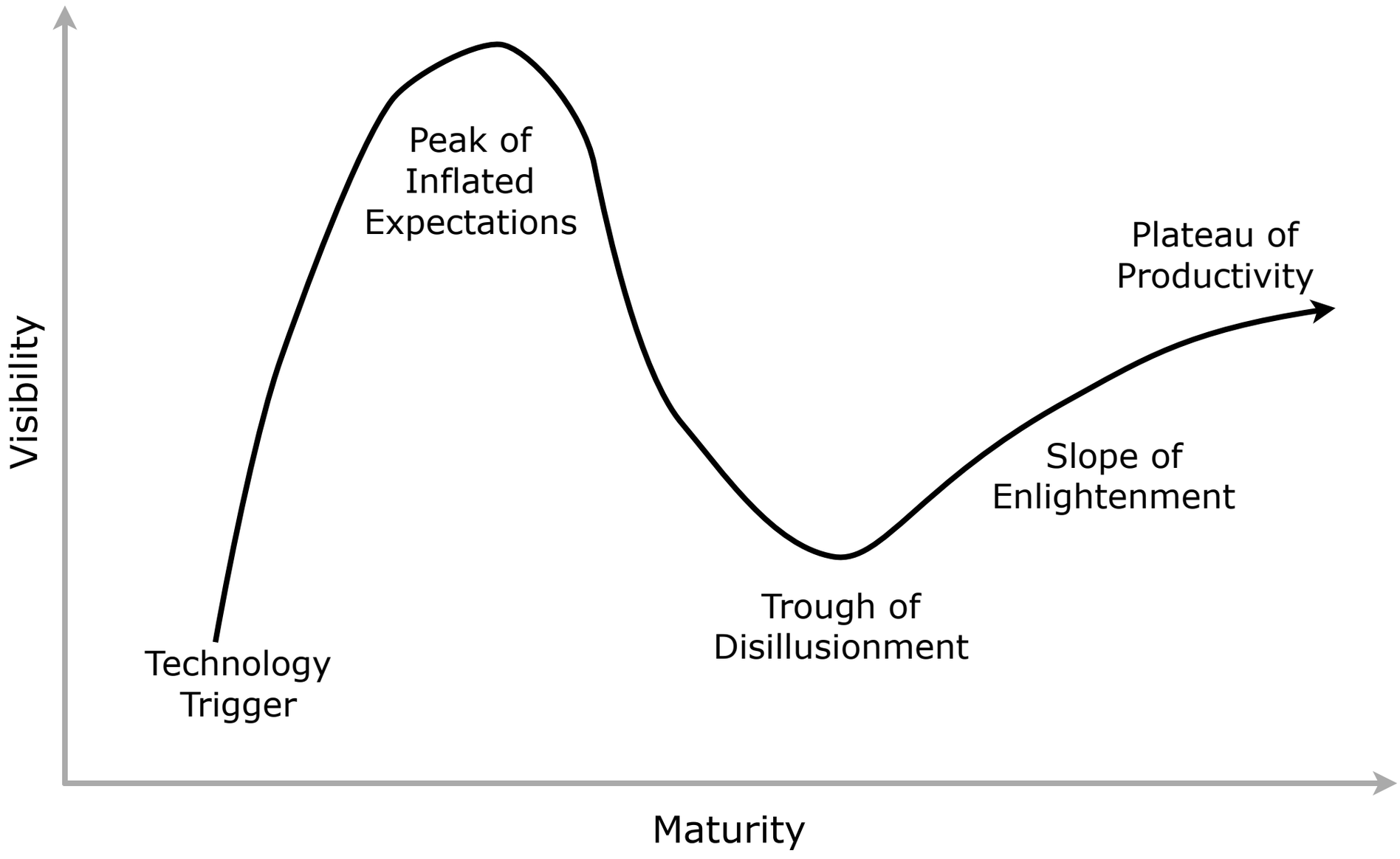



Space



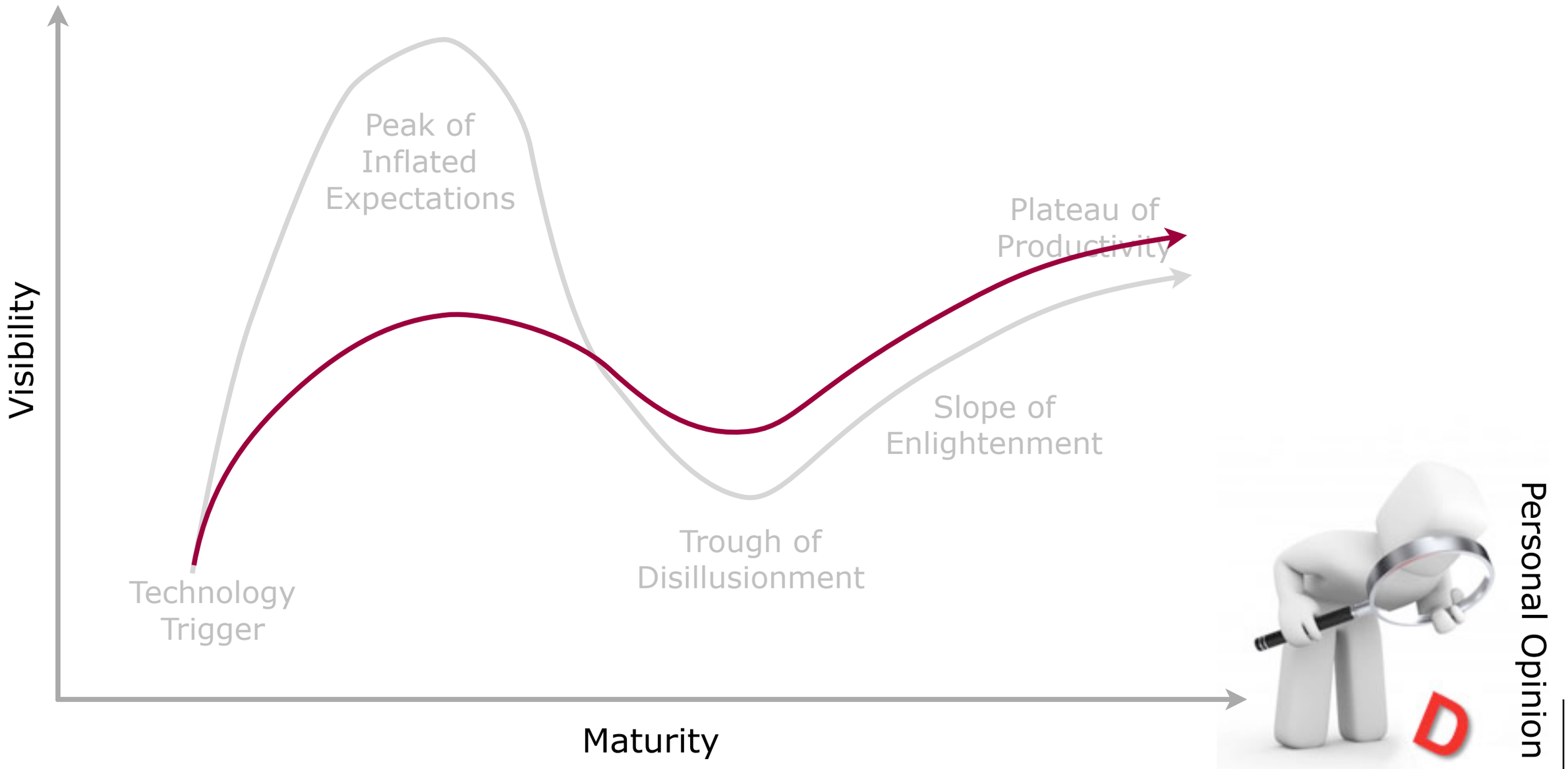
Place

# Hype Cycle



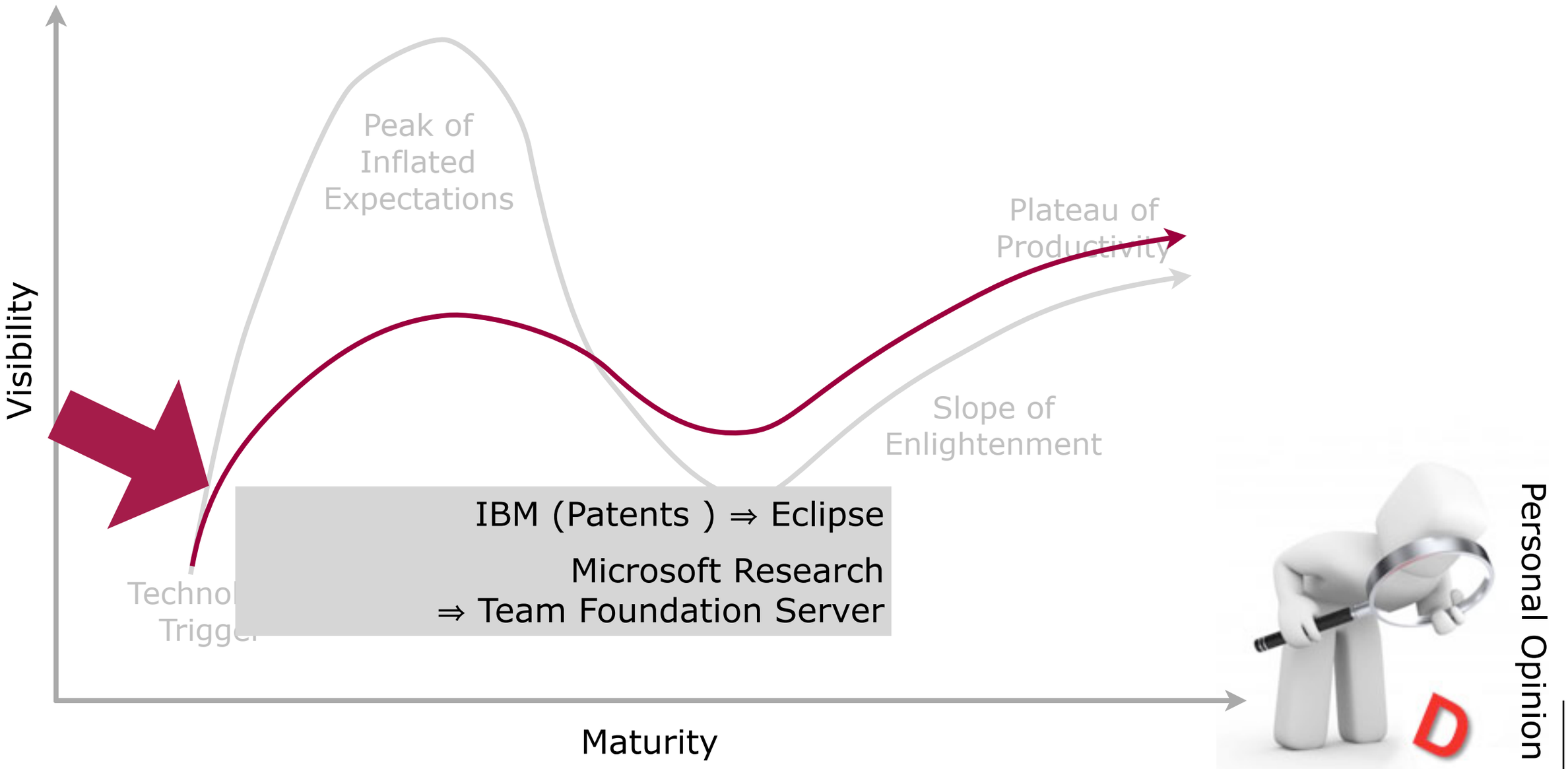

*Hype Cycle © Gartner*

# The Future ?




*Hype Cycle © Gartner*

# The Future ?




*Hype Cycle © Gartner*

# HOW ?


Changes
Past
annotations
recurring
program
Analyzing
Present
misclassified
Repeating
bug
links
re
long
Software
change-aware
reports
mining
transformations
traceability
meta
Future
Change-Centric
logging
meta-data
model
active
fix
evolution
API
process
distilling
monitor
bug-fixes
Development
Tracing
test


JIRA


GitHub


eclipse



Change
Database

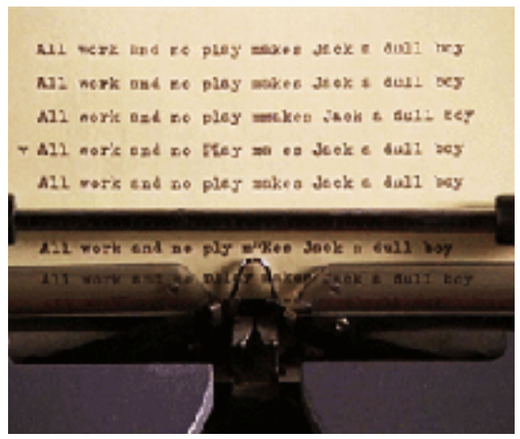
JIRA

GitHub

eclipse

Change
Database

Repetitive




Traceability

Estimation

Testing

# Project Plan

| | Year 1 | | | | Year 2 | | | | Year 3 | | | | Year 4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 | Q13 | Q14 | Q15 | Q16 |
| **WP0: Management** | | | | | | | | | | | | | | | | |
| steering board | 0.1.a | | | 0.1.b | | 0.1.c | | 0.1.d | | 0.1.e | | 0.1.f | | 0.1.g | | 0.1.h |
| IWT reporting | 0.2.a | | | 0.2.b | | | | 0.2.c | | | | 0.2.d | | | | 0.2.e |
| **WP1: State of the praxis versus state of the art** | | | | | | | | | | | | | | | | |
| site visits | 1.1.a | | | | | | | | 1.1.b | | | | | | | |
| tools comparison | | 1.2.a | | | | | | | | 1.2.b | | | | | | |
| **WP2: Analysing Change** | | | | | | | | | | | | | | | | |
| meta-model | 2.1.a | 2.1.b [P] | | | | | | | | | | | | 2.1.c | | 2.2.c |
| distilling & logging | | | | 2.2.a [P] | | | 2.2.b [P] | | | | | | | | | 2.2.c |
| **WP3: Repeating Changes** | | | | | | | | | | | | | | | | |
| transformations | | | 3.1.a | | 3.1.b [P] | | | | | | | | | | | |
| change-aware | | | | | | 3.2.a [P] | | | | | | 3.2.b [P] | | | | 3.2.c |
| **WP4: Tracing Changes** | | | | | | | | | | | | | | | | |
| traceability links | | | | | | 4.1.a [P] | | | | | | | | | | 4.1.b |
| annotation | | | | | | | 4.2.a [P] | | | | | | | | | 4.2.b |
| **WP5: Valorisation activities** | | | | | | | | | | | | | | | | |
| pilot cases | | | | | 5.1.a | | | 5.1.b | | 5.1.c | | | 5.1.d | | | 5.1.e |
| dissemination evt. | | | | | | | | 5.2.a | | | | | | | | 5.2.b |

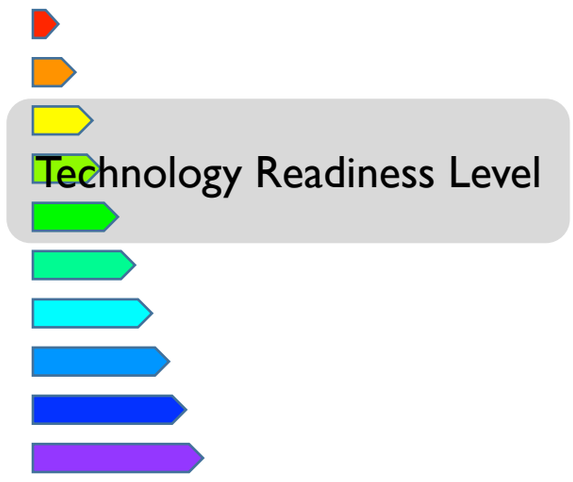
**m.n.x** Deliverable m.n.x is due at end of quarter. m = work-package; n = activity; x = sequence number
**m.n.x [P]** Deliverable m.n.x includes a prototype tool

# Project Plan

| | Year 1 | | | | Year 2 | | | | Year 3 | | | | Year 4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 | Q13 | Q14 | Q15 | Q16 |
| WP0: Management | | | | | | | | | | | | | | | | |
| steering board | 0.1.a | | | 0.1.b | | 0.1.c | | 0.1.d | | 0.1.e | | 0.1.f | | 0.1.g | | 0.1.h |
| IWT reporting | 0.2.a | | | 0.2.b | | | | 0.2.c | | | | 0.2.d | | | | 0.2.e |
| WP1: State of the praxis versus state of the art | | | | | | | | | | | | | | | | |
| site visits | 1.1.a | | | | | | | | 1.1.b | | | | | | | |
| tools comparison | | 1.2.a | | | | | | | | 1.2.b | | | | | | |
| WP2: Analysing Change | | | | | | | | | | | | | | | | |
| meta-model | 2.1.a | 2.1.b [P] | | | | | | | | | | | | 2.1.c | | 2.2.c |
| distilling & logging | | | | 2.2.a [P] | | | 2.2.b [P] | | | | | | | | | 2.2.c |
| WP3: Repeating Changes | | | | | | | | | | | | | | | | |
| transformations | | | 3.1.a | | 3.1.b [P] | | | | | | | | | | | |
| change-aware | | | | | | 3.2.a [P] | | | | | | 3.2.b [P] | | | | 3.2.c |
| WP4: Tracing Changes | | | | | | | | | | | | | | | | |
| traceability links | | | | | | 4.1.a [P] | | | | | | | | | | 4.1.b |
| annotation | | | | | | | 4.2.a [P] | | | | | | | | | 4.2.b |
| WP5: Valorisation activities | | | | | | | | | | | | | | | | |
| pilot cases | | | | | 5.1.a | | | 5.1.b | | | 5.1.c | | | 5.1.d | | 5.1.e |
| dissemination evt. | | | | | | | | 5.2.a | | | | | | | | 5.2.b |

**m.n.x** Deliverable m.n.x is due at end of quarter. m = work-package; n = activity; x = sequence number
**m.n.x [P]** Deliverable m.n.x includes a prototype tool

# Pilot Cases
# Dissemination Event




join ?


Changes
Past
annotatio
recurring
program
Analyzing
Present
misclassified
Software
change-aware
reports
mining
Repeating
bug
links
relong
transformations
traceability
meta
Future
Change-Centric
Tracing
test
logging
meta-data
model
active
fix
evolution
API
process
distilling
monitor
bug-fixes
Development

# Technology Readiness Level

AGENTSCHAP INNOVEREN & ONDERNEMEN

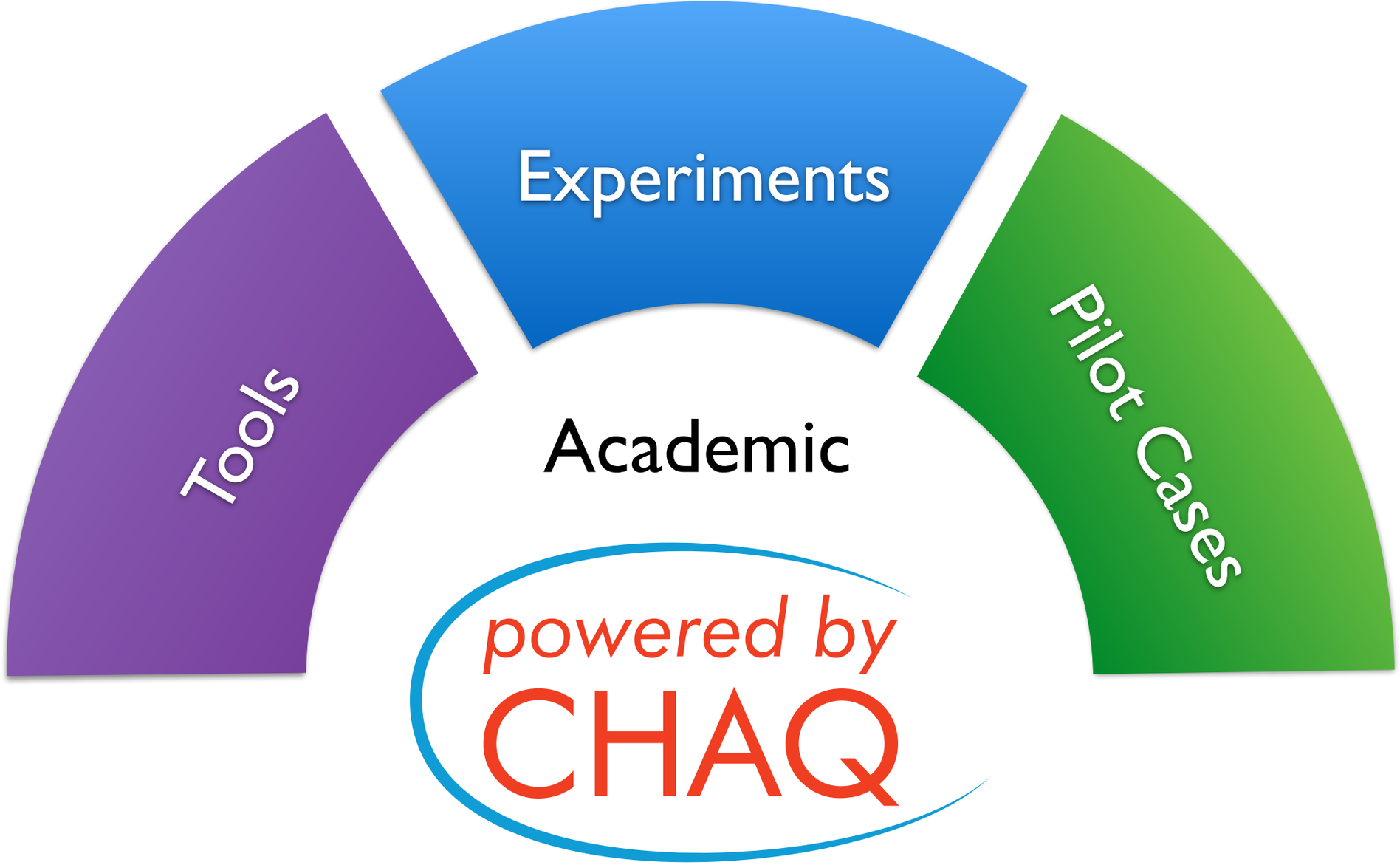
- **TRL 0** — *Idea.* Unproven concept, no testing has been performed
- **TRL 1** — *Basic research.* Principles postulated and observed; no experimental proof available
- **TRL 2** — *Technology formulation.* Concept and application have been formulated
- **TRL 3** — *Applied research.* First laboratory tests completed; proof of concept
- **TRL 4** — *Small scale prototype* built in a laboratory environment ("ugly" prototype)
- **TRL 5** — *Large scale prototype tested* in intended environment
- **TRL 6** — *Prototype system tested* in intended environment close to expected performance
- **TRL 7** — *Demonstration system* operating in operational environment at pre-commercial scale
- **TRL 8** — *First of a kind commercial* system. Manufacturing issues solved
- **TRL 9** — *Full commercial application*, technology available for consumers.

# Technology Readiness Level

AGENTSCHAP INNOVEREN & ONDERNEMEN

- **TRL 0** — *Idea.* Unproven concept, no testing has been performed
- **TRL 1** — *Basic research.* Principles postulated and observed; no experimental proof available
- **TRL 2** — *Technology formulation.* Concept and application have been formulated
- **TRL 3** — *Applied research.* First laboratory tests completed; proof of concept
- **TRL 4** — *Small scale prototype* built in a laboratory environment ("ugly" prototype)
- **TRL 5** — *Large scale prototype tested* in intended environment
- **TRL 6** — *Prototype system tested* in intended environment close to expected performance
- **TRL 7** — *Demonstration system* operating in operational environment at pre-commercial scale
- **TRL 8** — *First of a kind commercial* system. Manufacturing issues solved
- **TRL 9** — *Full commercial application*, technology available for consumers.


powered by
CHAQ


Experiments
Tools
Pilot Cases
Academic
powered by
CHAQ


Tools
Experiments
Pilot Cases
Academic
powered by
CHAQ
Consultants
Industrial
Tool Vendors
Software
Intensive
Products