

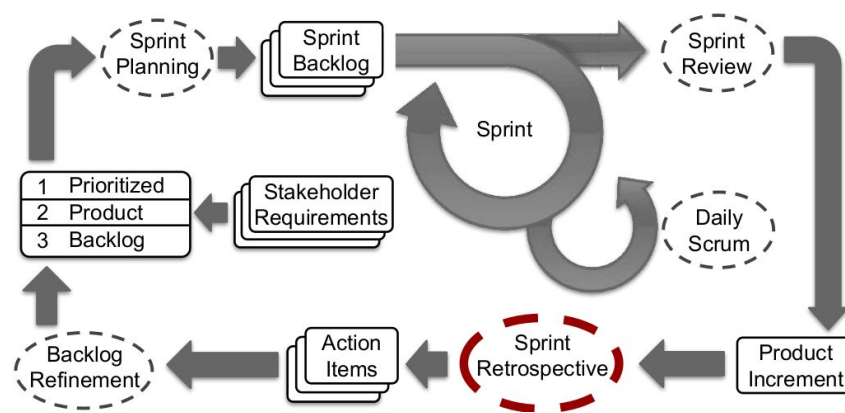
Playing With Your Project Data in Scrum Retrospectives

Christoph Matthies, Hasso Plattner Institute, University of Potsdam, Germany

Scrum & Retrospectives

- Retrospectives: **Agile's process improvement approach**
- They should "start with the hard data" [1]
- ➔ However, programmers' beliefs primarily formed on personal experience rather than **empirical project evidence** [2]

Kniberg: "retro is the number-one-most-important thing in Scrum" [3]



Retrospective Games

- Activities to structure Retros & encourage sharing impressions [1]
- Most popular resource lists 139 activities, 36 to *gather data* [4]

Only **2/36** proposed *gather data* activities take project data into account!

Software Project Data

- Software engineers **continuously document their work**, e.g. work item descriptions, test logs, and code commits.
- Employed development processes are "**inscribed**" into project artifacts [5]
- ➔ Provides **evidence for problems**, e.g. build failures, bug reports

Project Data Activities

- Actively use (& play with) own project data in Retrospective meetings
- Learn from last iteration's project data** for the next iteration
- Use tools that devs are already familiar with, e.g. *git*
- Devs are experts on their own data ➔ **interpretable results!**
- Start discussions on improvements** based on team data

Example: Progress Check

- Issue: Single dev commits all code
- Train team in VCS usage
- Estimate the improvement
- Track contributor count using `git shortlog -sn|wc -l`

Data Source	Example Measurement
Version Control	Regular, small code commits
Issue Tracker	Equal distribution of work items
Software Tests	Regularity of builds, failures
Status Monitor	Accumulated downtime
Comm. Tools	Communication patterns
Code Analysis	Development of code coverage

➔ Contrast **mental process model** of Retrospective participants and their **own project data realities!**

