Lean Software Development. Issues in Agile. Week 7

Giancarlo Succi, Artem Kruglov

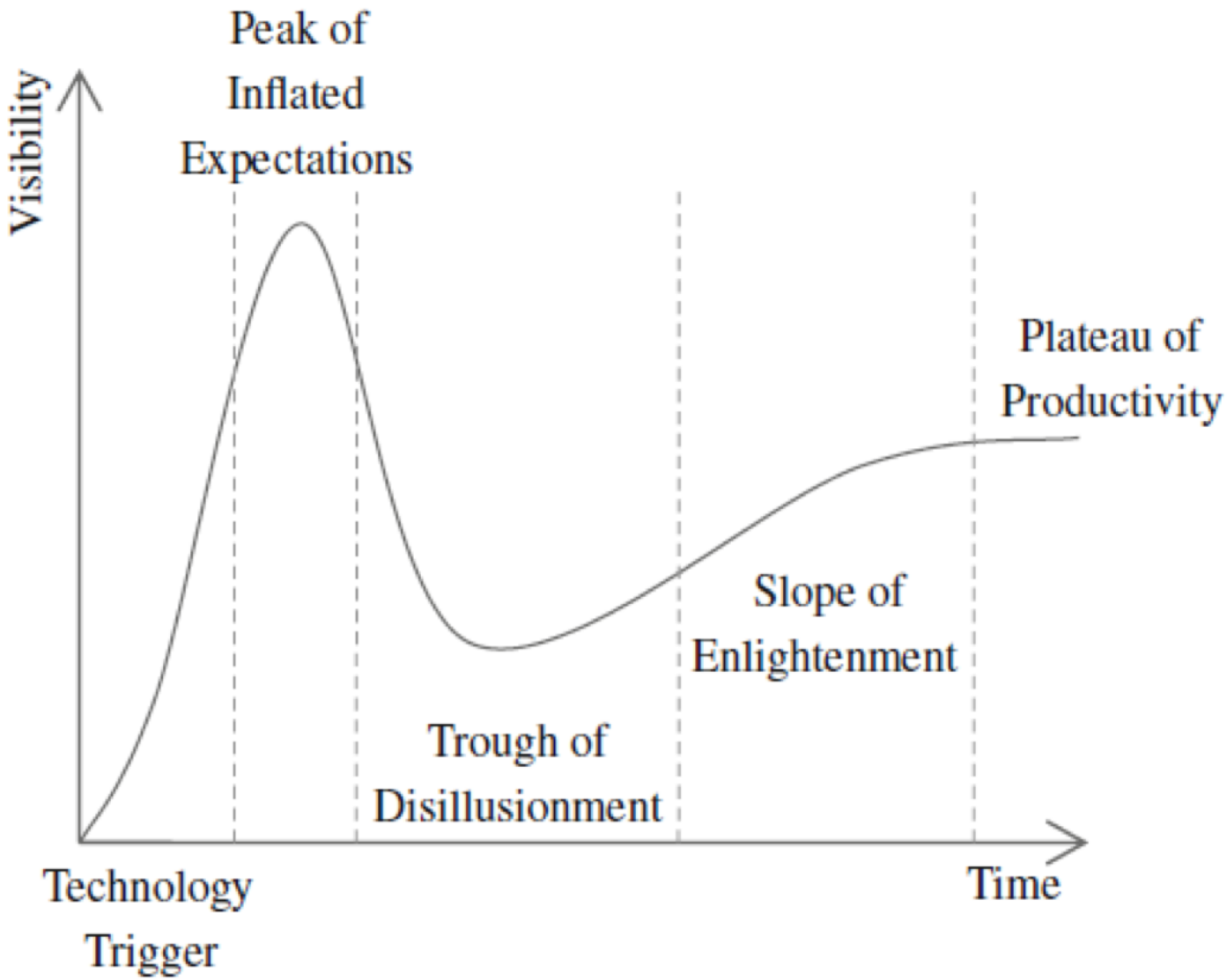
Innopolis University

January 12, 2023

Outline

- **Gartner's Innovation Hype Cycle**
- The Dark Side of Agile
- The Skepticism Towards Agile Methods
- The Zen of Agile
- Summary

Gartner's Innovation Hype Cycle



Gartner's Innovation Hype Cycle

1. The Technology Trigger.

A potential technology breakthrough kicks things off. Early proof-of-concept stories and media interest trigger significant publicity. Often no usable products exist and commercial viability is unproven.

[1] https://en.wikipedia.org/wiki/Hype_cycle

2. The Peak of Inflated Expectations.

Early publicity produces a number of success stories – often accompanied by scores of failures. Some companies take action; most don't.

[1] https://en.wikipedia.org/wiki/Hype_cycle

3. The Trough of Disillusionment.

Interest wanes as experiments and implementations fail to deliver. Producers of the technology shake out or fail. Investments continue only if the surviving providers improve their products to the satisfaction of early adopters.

[1] `https://en.wikipedia.org/wiki/Hype_cycle`

4. The Slope of Enlightenment.

More instances of how the technology can benefit the enterprise start to crystallize and become more widely understood. Second- and third-generation products appear from technology providers. More enterprises fund pilots; conservative companies remain cautious.

[1] https://en.wikipedia.org/wiki/Hype_cycle

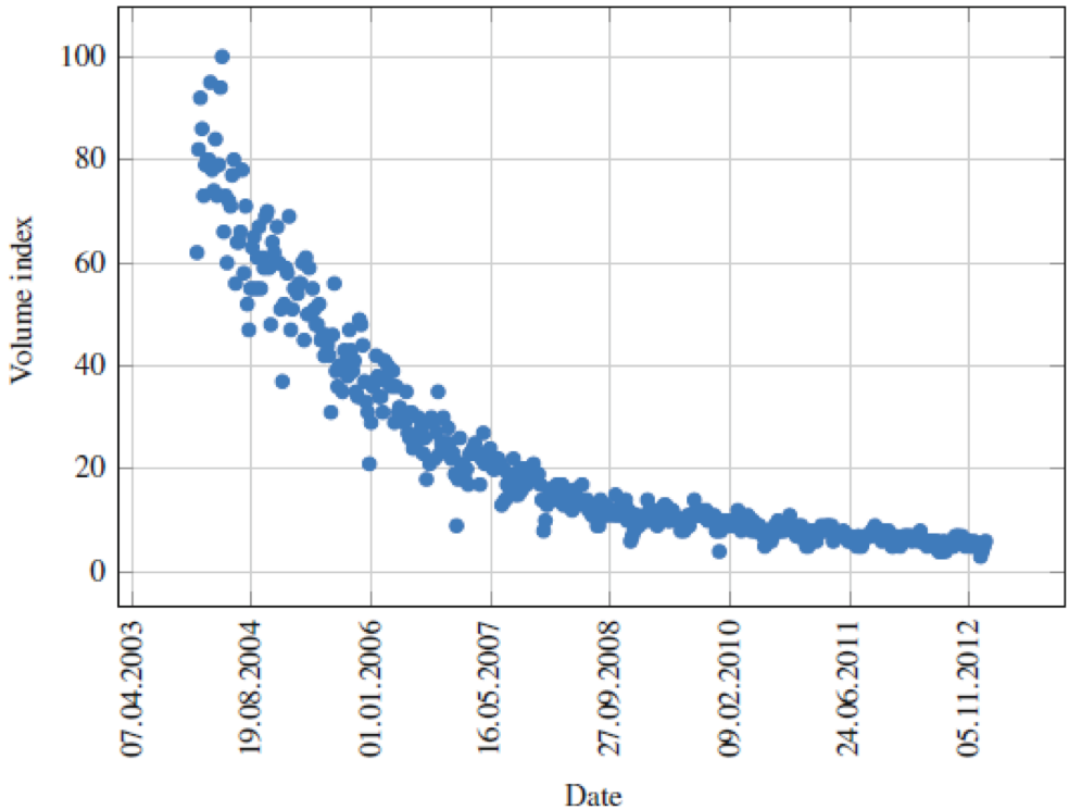
5. The Plateau of Productivity.

Mainstream adoption starts to take off. Criteria for assessing provider viability are more clearly defined. The technology's broad market applicability and relevance are clearly paying off.

[1] https://en.wikipedia.org/wiki/Hype_cycle

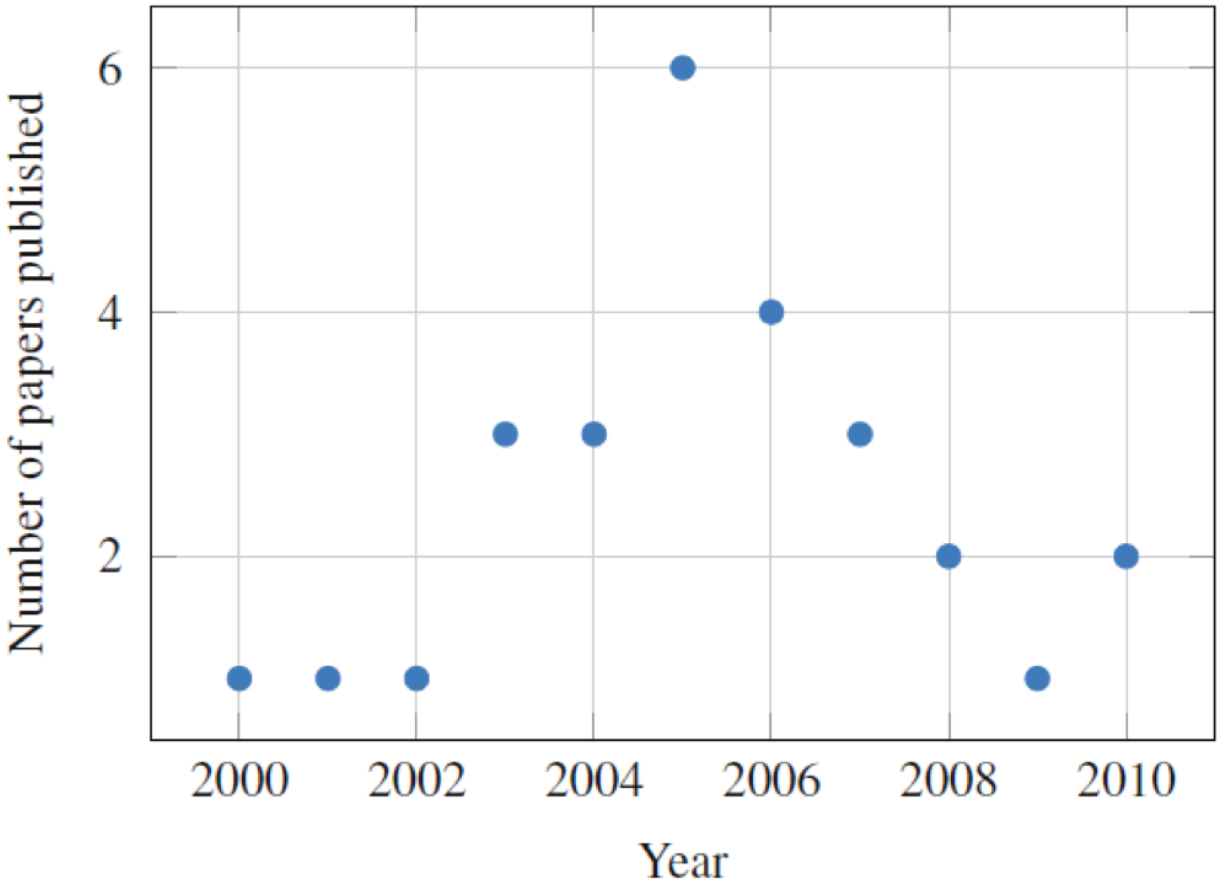
Gartner's Innovation Hype Cycle

Extreme Programming popularity has followed quite closely the Gartner's Hype Cycle: after the first hype, there has been a period of disillusion



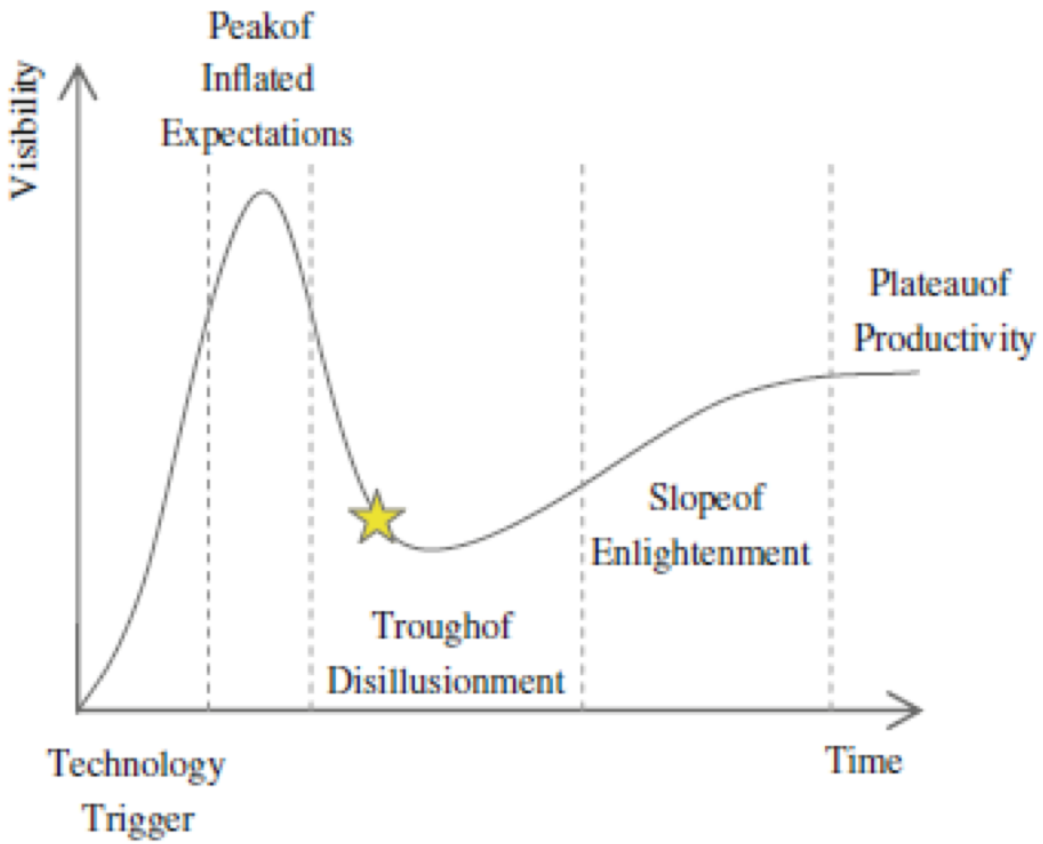
Gartner's Innovation Hype Cycle

Published articles containing the terms "Agile," "extreme programming," or "scrum" in their title or abstract on the International Conference on Software Engineering from 2000 to 2010



Gartner's Innovation Hype Cycle

Where the Gartner Group sees the "Project-Oriented Agile Development Methodology" now



Outline

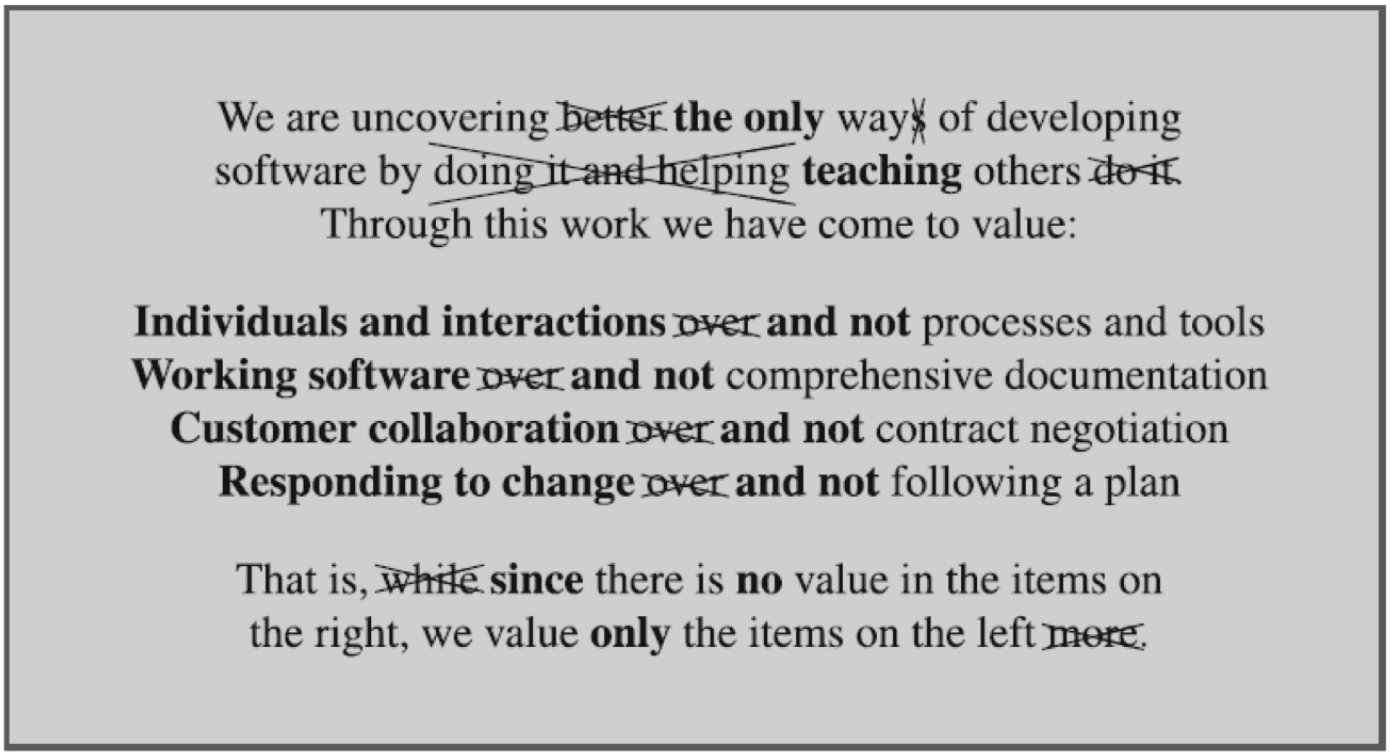
- Gartner's Innovation Hype Cycle
- **The Dark Side of Agile**
- The Skepticism Towards Agile Methods
- The Zen of Agile
- Summary

The Dark Side of Agile

Agility was interpreted just as the cut in complexity, a rather simplistic and unjustified cut in complexity whatever the term complexity was—documentation, good development practices as design, etc. (continued)

The Dark Side of Agile

Enthusiastic early subscribers to the Agile Manifesto became zealot. So they read the Agile manifesto as in Figure



The Dark Side of Agile

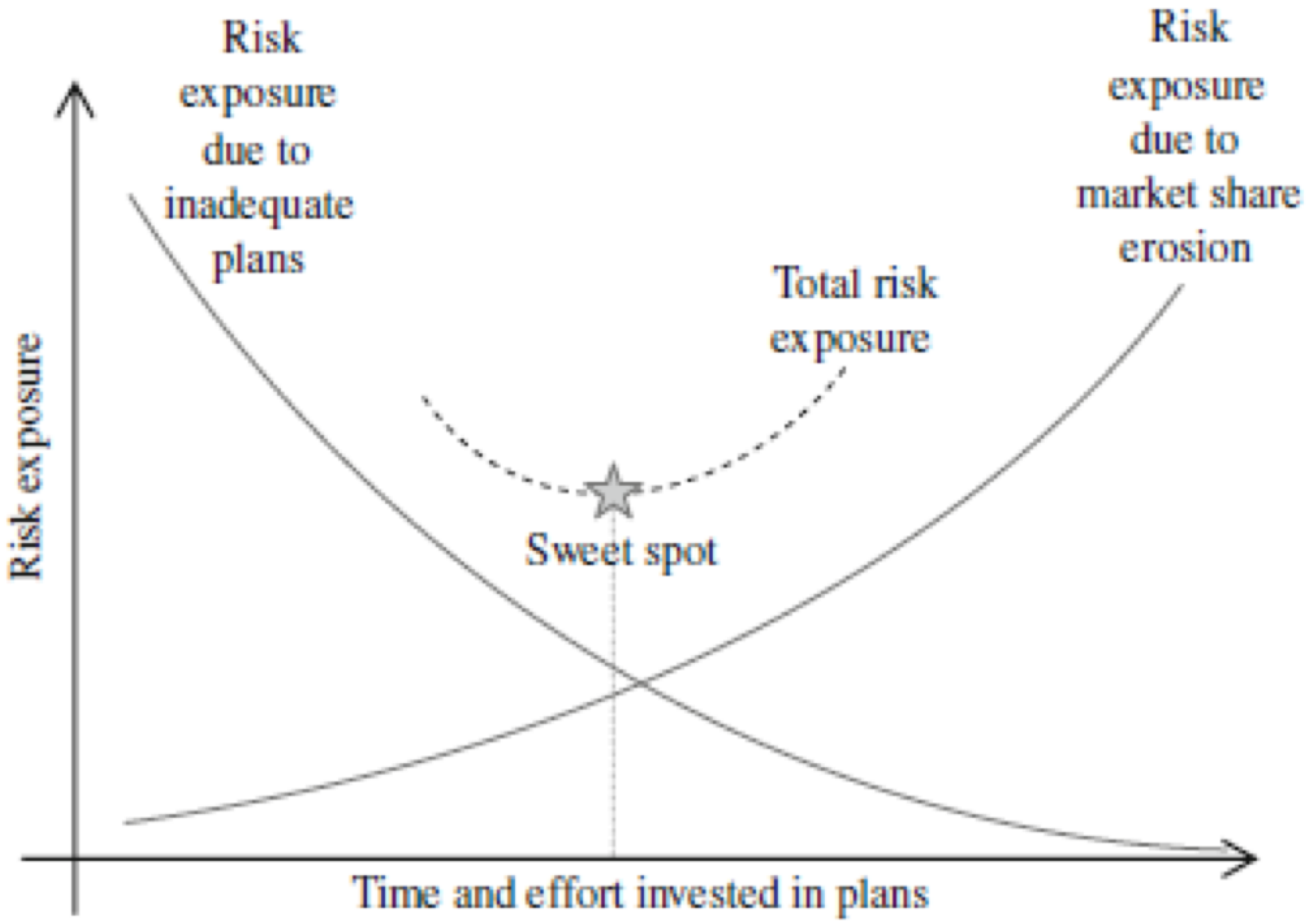
In fact, the misconceptions about Agile can be linked to the four statements of the "Dark Agile Manifesto":

- Individuals and interactions over processes and tools: "Talking to people instead of using a process gives us the freedom to do whatever we want."
- Working software over comprehensive documentation: "We want to spend all our time coding. Remember, real programmers don't write documentation."
  (continued)

The Dark Side of Agile

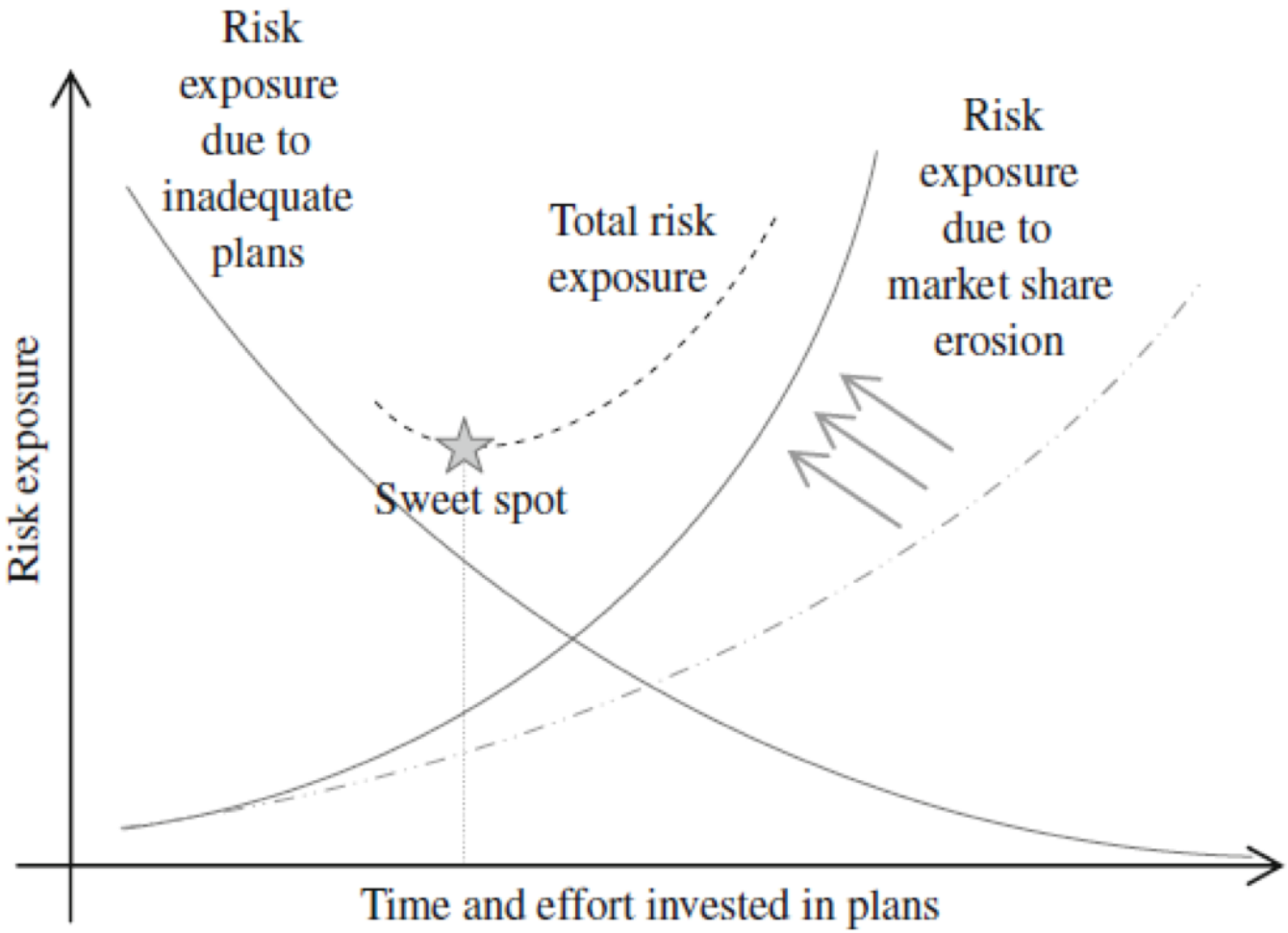
- Customer collaboration over contract negotiation: “Haggling over the details is merely a distraction from the real work of coding. We’ll work out the details once we deliver something.”
- Responding to change over following a plan: “Following a plan implies we have to think about the problem and how we might actually solve it. Why would we want to do that when we could be coding?”

The Dark Side of Agile



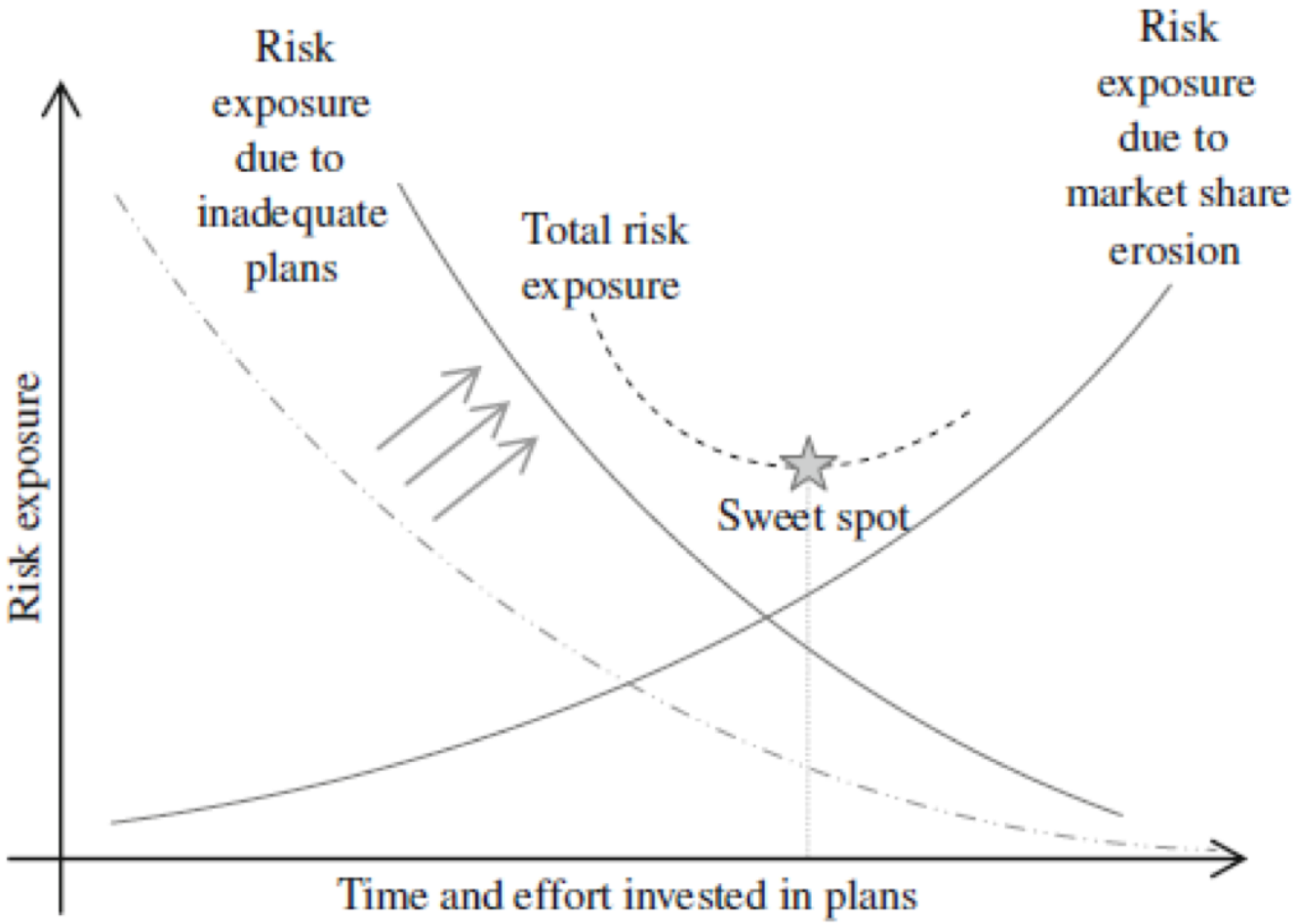
Determination of the sweet spot

The Dark Side of Agile



Sweet spot in companies in turbulent markets

The Dark Side of Agile



Sweet spot in companies in markets requiring thorough planning

Outline

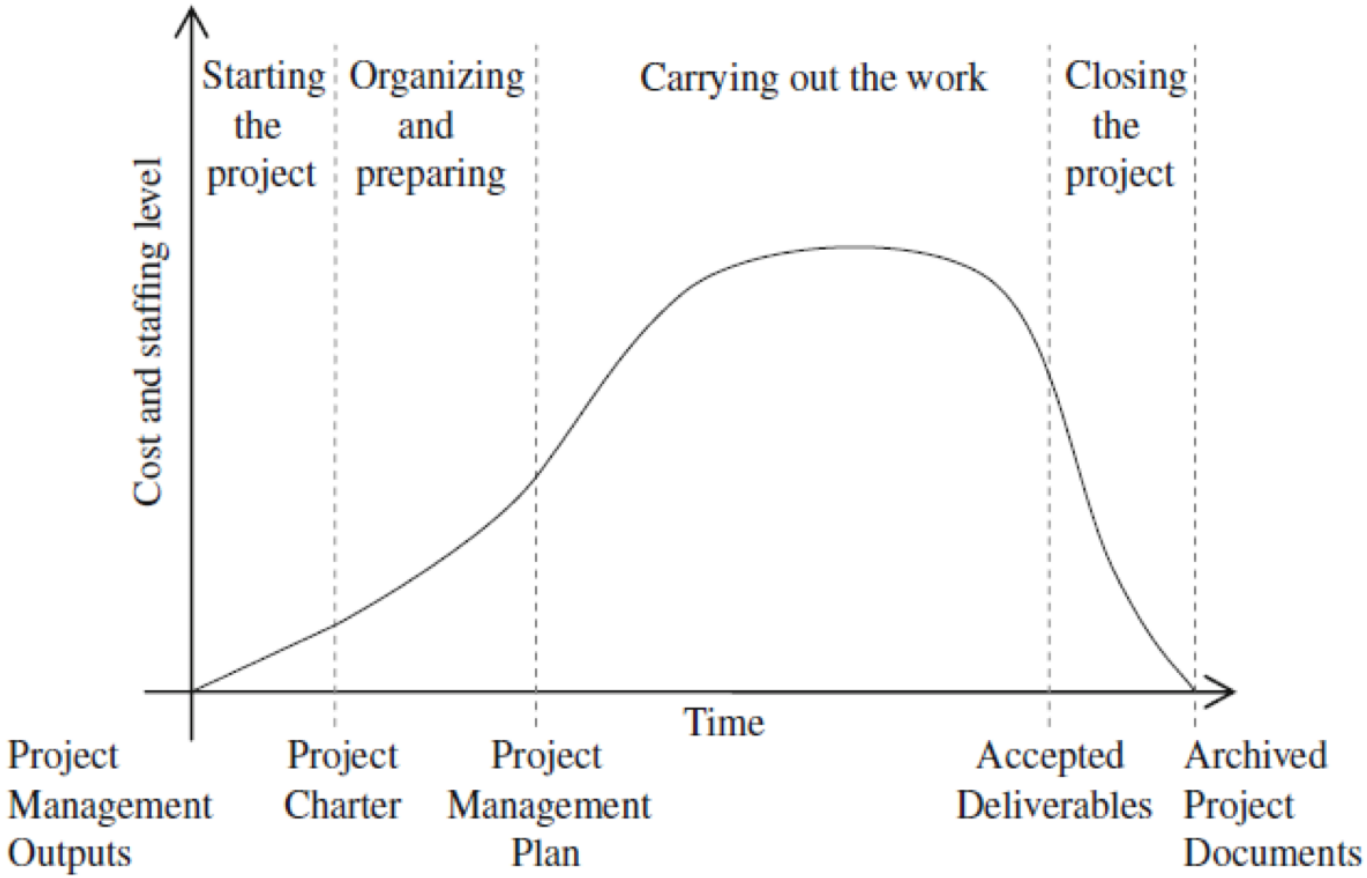
- Gartner's Innovation Hype Cycle
- The Dark Side of Agile
- **The Skepticism Towards Agile Methods**
- The Zen of Agile
- Summary

The Skepticism Towards Agile Methods

The adaptive nature of Agile methods was conceived with a premise in mind: it is not possible to plan every detail in advance. This goes against conventional wisdom, which teaches us to "first think, then do."

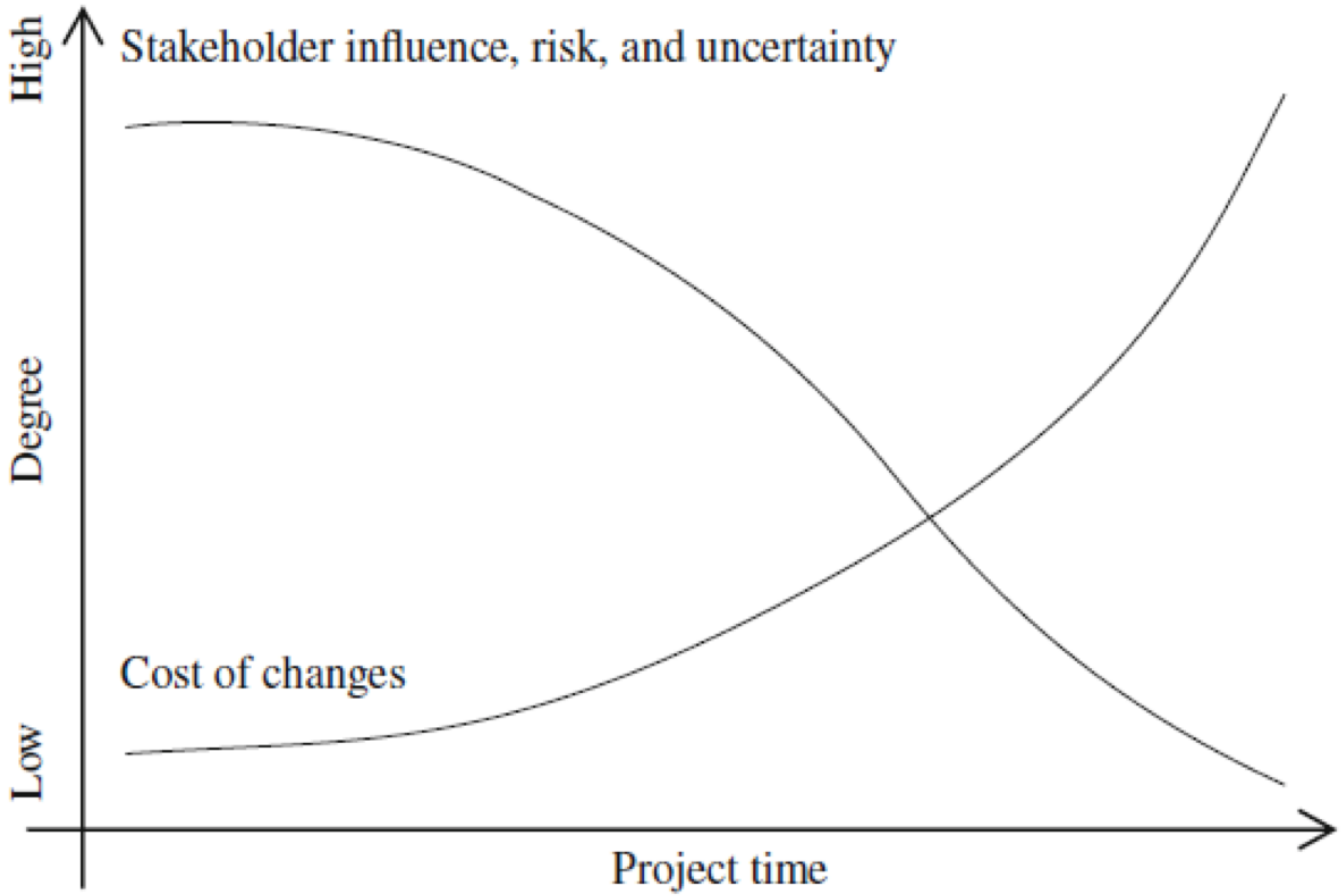
Agile software development goes against the current best practices of project management. Software development projects are seen as any other project, and therefore, project managers assume they should be managed as any other project.

The Skepticism Towards Agile Methods



Typical cost and staffing levels across the project life cycle (waterfall model)

The Skepticism Towards Agile Methods



Stakeholder influence, risk, uncertainty, and the cost of changes during the project (waterfall model)

The Skepticism Towards Agile Methods

Managers without IT background expect to work like the PMBOK standard teaches and will oppose an approach that appears to:

- Promote inadequate preparation.
- Accept exploding costs of changes.
- Accept a high risk.

The Skepticism Towards Agile Methods

To put it in other words, if one looks at Agile methods from the perspective of the Guide to the Project Management Body of Knowledge, the conclusions may be wrong because they are based on assumptions that differ from the assumptions of Agile methods.

The Skepticism Towards Agile Methods

To overcome these differences it is important to agree on:

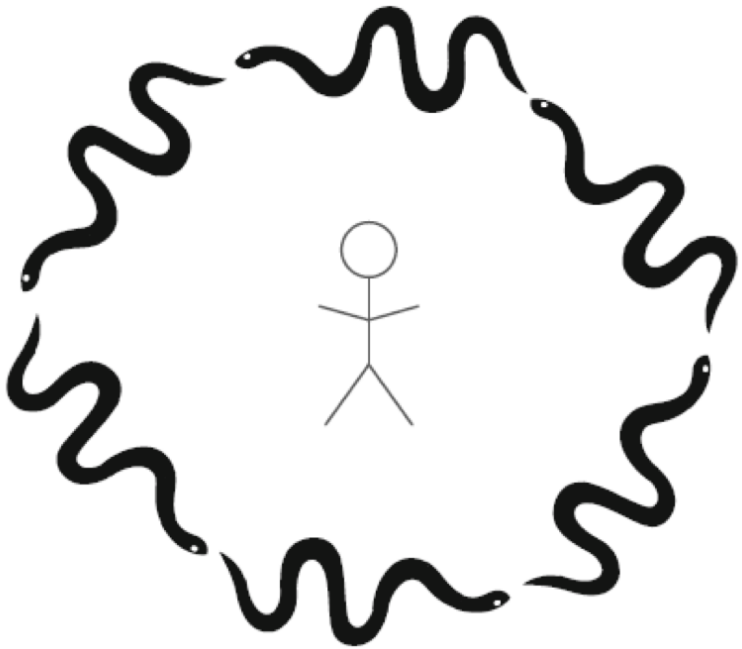
- "Where we are": an assessment of the current situation.
- "Where we want to go": a description of the goals.
- "How to get there": a description of the method to achieve the goal.

Outline

- Gartner's Innovation Hype Cycle
- The Dark Side of Agile
- The Skepticism Towards Agile Methods
- **The Zen of Agile**
- Summary

The Zen of Agile

Stephens and Rosenberg compare Extreme Programming with a ring of poisonous snakes, daisy-chained together. Each snake represents a practice that has issues that are compensated by the use of another practice. (continued)



The Zen of Agile

They show that the different practices depend on each other such as:

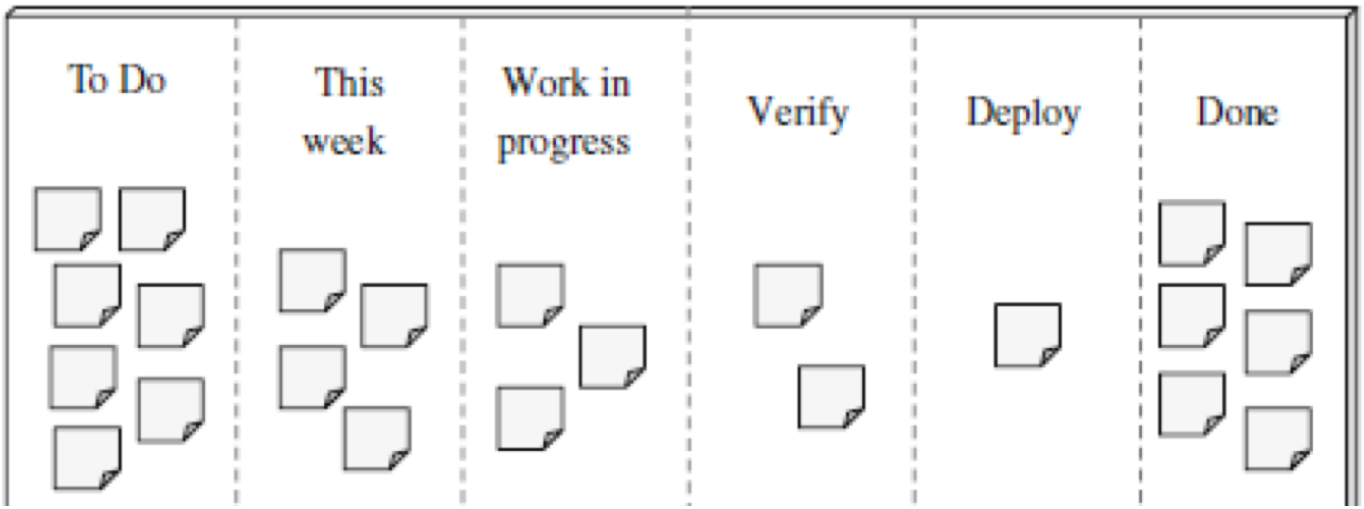
- Requirements are elicited gradually, which – since developers do not have a complete picture – can result in wasted effort. This problem is alleviated by incremental design, i.e., to be ready to change the design incrementally.
- Incremental design avoids designing everything upfront to minimize the risk of being wrong. This practice is considered safe because the code is constantly refactored.
  (continued)

The Zen of Agile

- Constant refactoring involves rewriting existing code that was thought to be finished: it went through discussions and thinking to get the design right. It potentially introduces new defects, but it is considered safe because of the presence of unit tests.
- Unit tests act as a safety net for incremental design and constant refactoring.With unit tests it is difficult to verify the quality of the design of an architecture. Pair programming alleviates this problem.

The Zen of Agile

Extreme Programming promotes the “Informative workspace” practice, and uses a task board to visualize the current status of the project. Usually divided into different sections that describe the status of each work station, user stories are placed on the task board depending on their status. (continued)



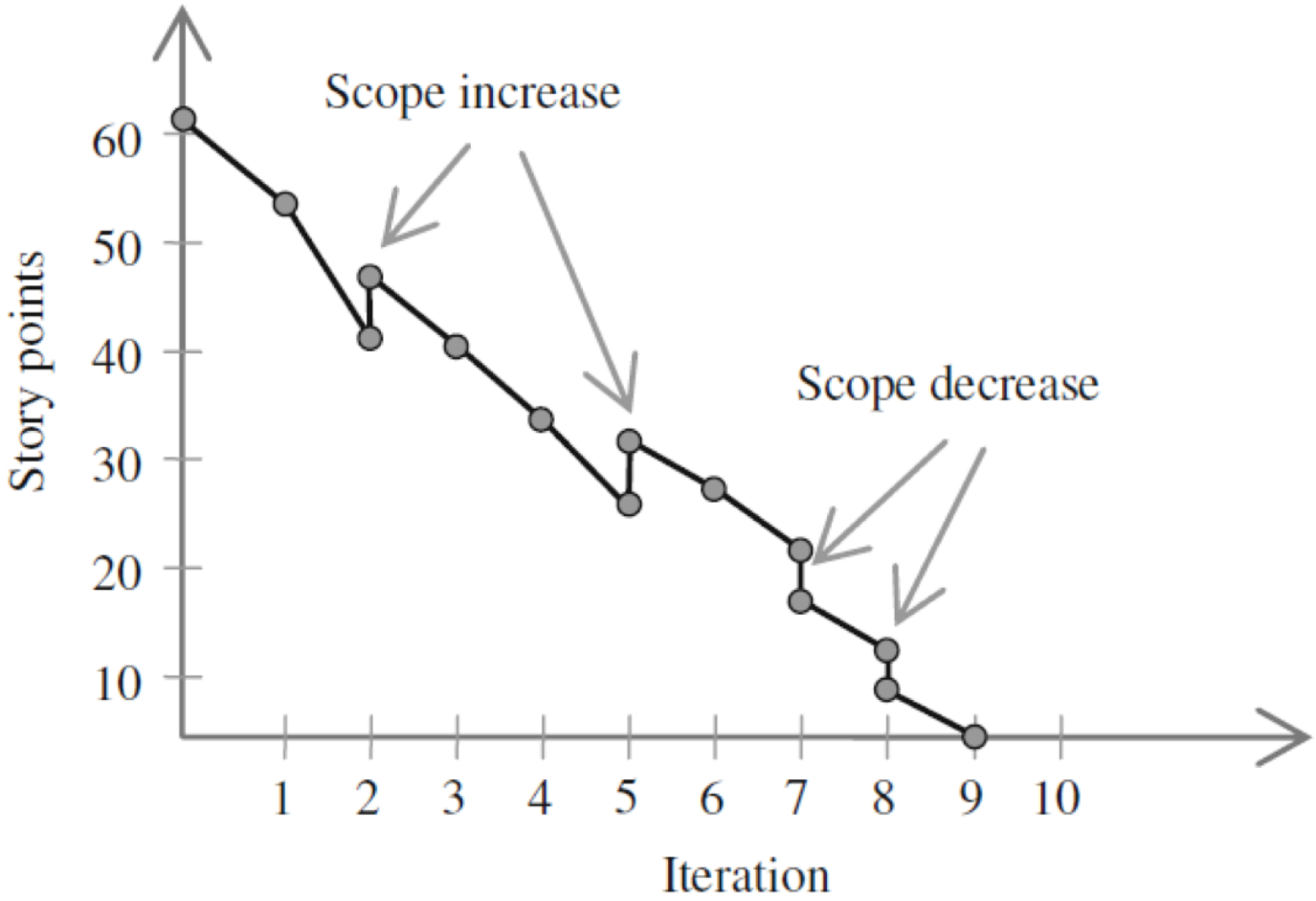
The Zen of Agile

The task board embodies the knowledge that:

- a task can only pass through a predefined set of phases;
- the first phase is the leftmost;
- the last phase is the rightmost;
- the sequence of phases is defined as the phases from left to right;
- many tasks stuck in the same phase signal a problem (the space for one phase is limited).

The Zen of Agile

Scrum or Crystal Clear use “burn-down charts”, charts that show the development progress against the predictions.



Outline

- Gartner's Innovation Hype Cycle
- The Dark Side of Agile
- The Skepticism Towards Agile Methods
- The Zen of Agile
- **Summary**