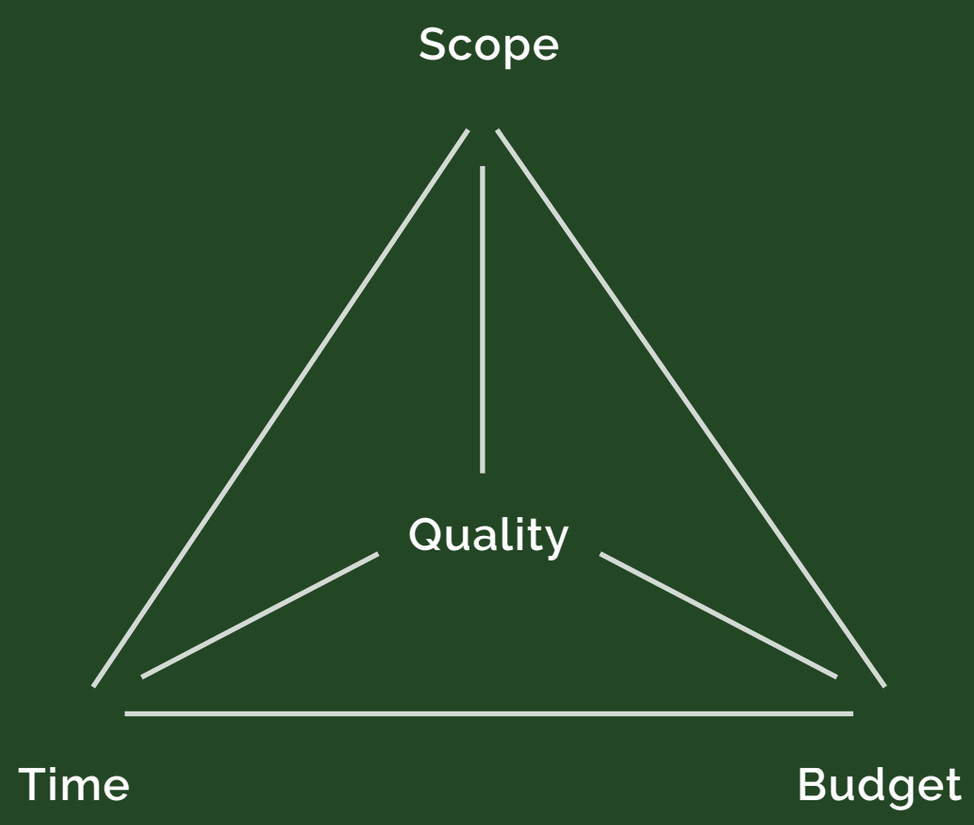
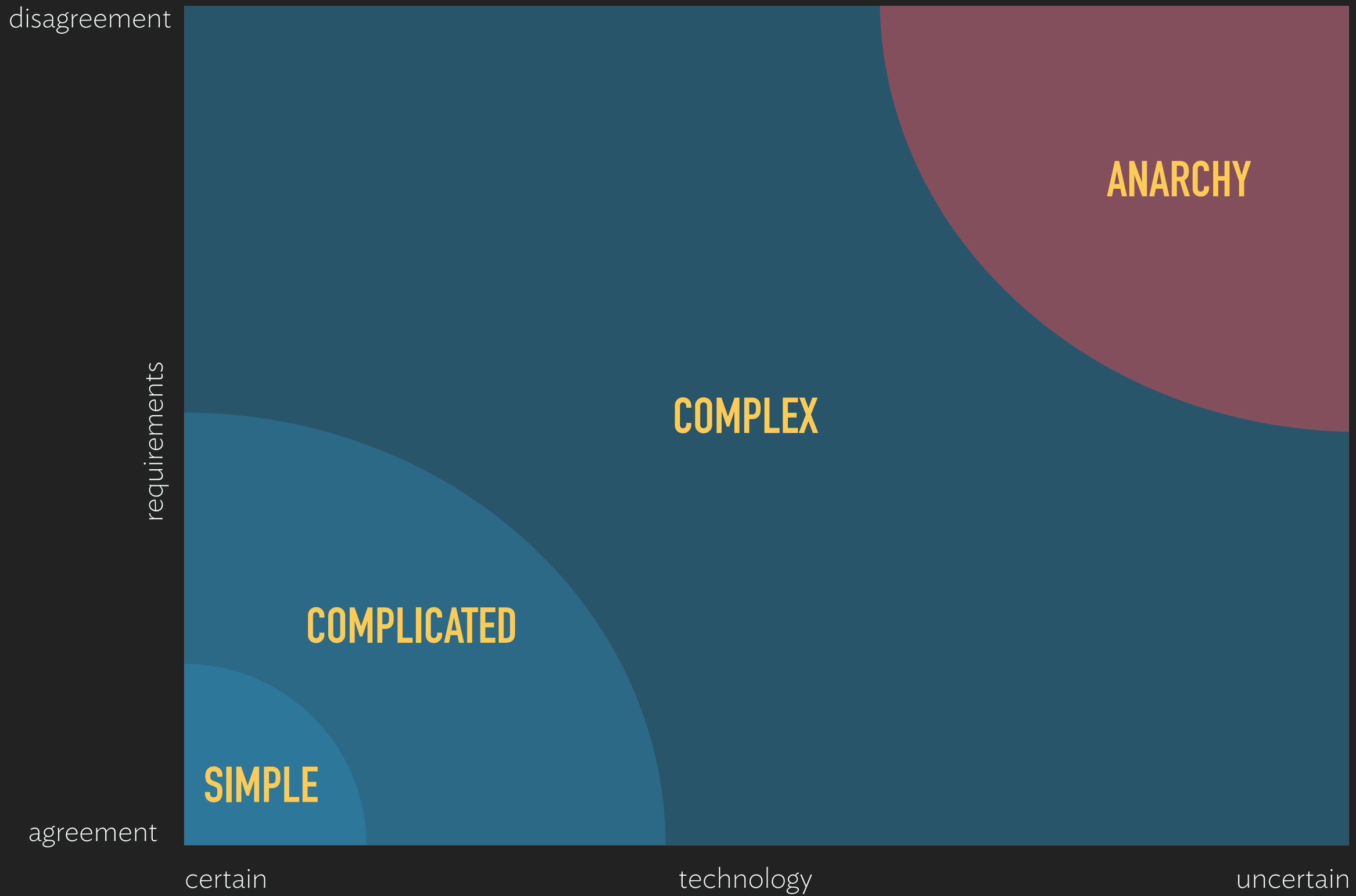
PRINCIPLES OF AGILE MANAGEMENT

SOFTWARE DEVELOPMENT LABORATORY

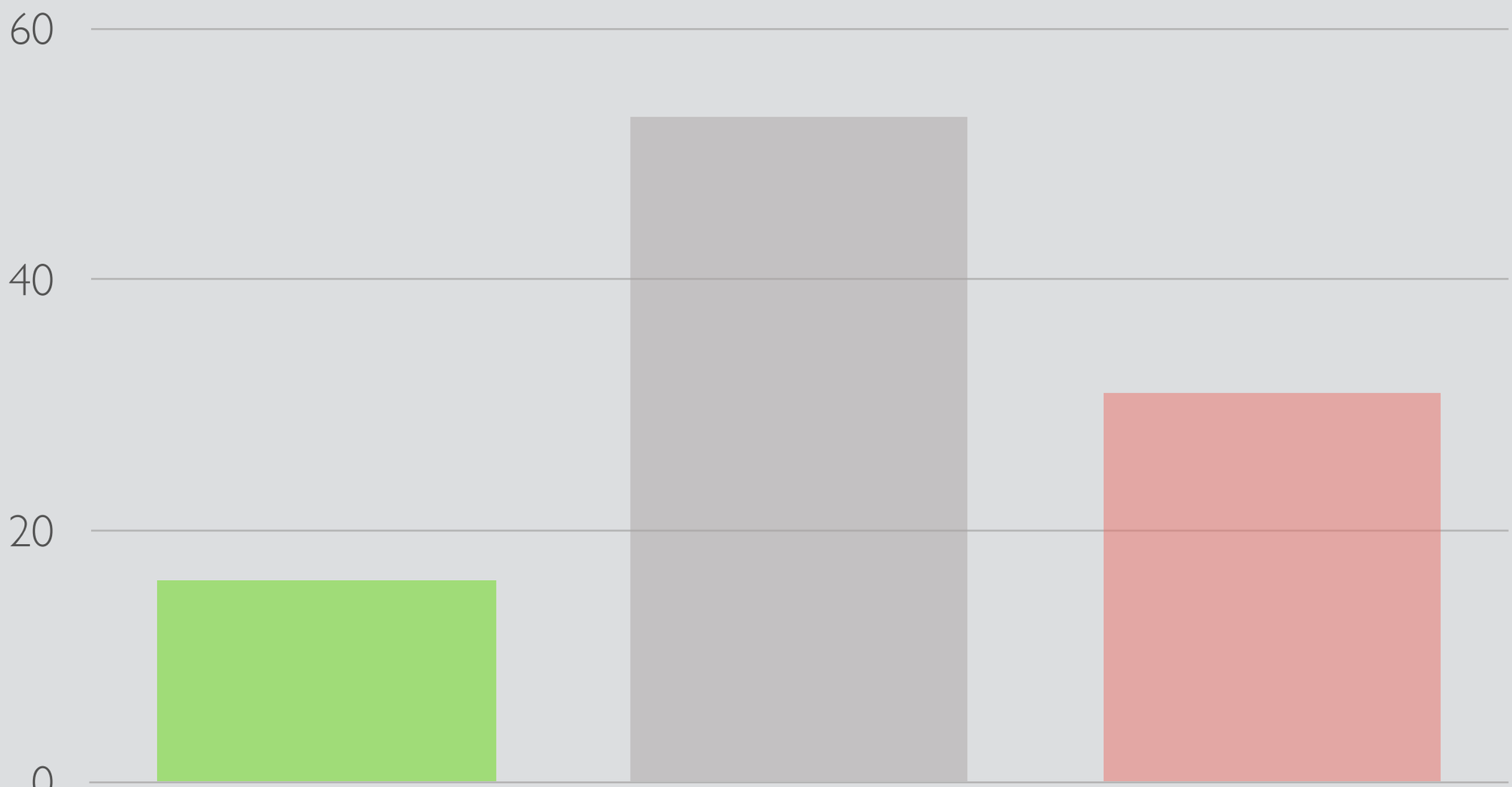
CONTROLLING PROJECTS



Where **time** is fixed and exactly the same **for all your projects**!

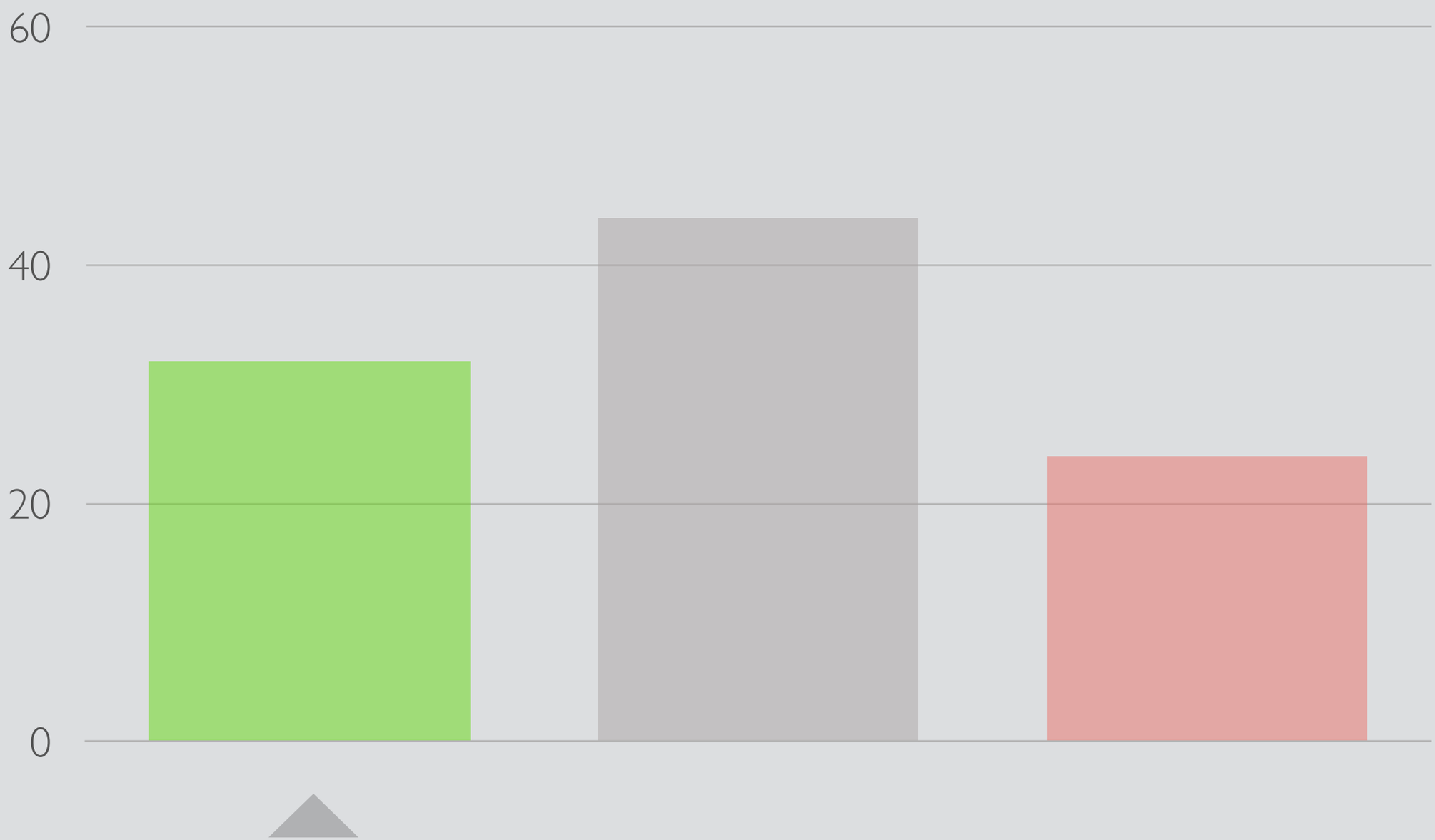

disagreement
requirements
ANARCHY
COMPLEX
COMPLICATED
SIMPLE
agreement
certain
technology
uncertain

SOFTWARE PROJECTS, CIRCA 1994



{ standish group international, the chaos report }

SOFTWARE PROJECTS, CIRCA 2009



That's still a success rate lower than 2 in 5 projects!

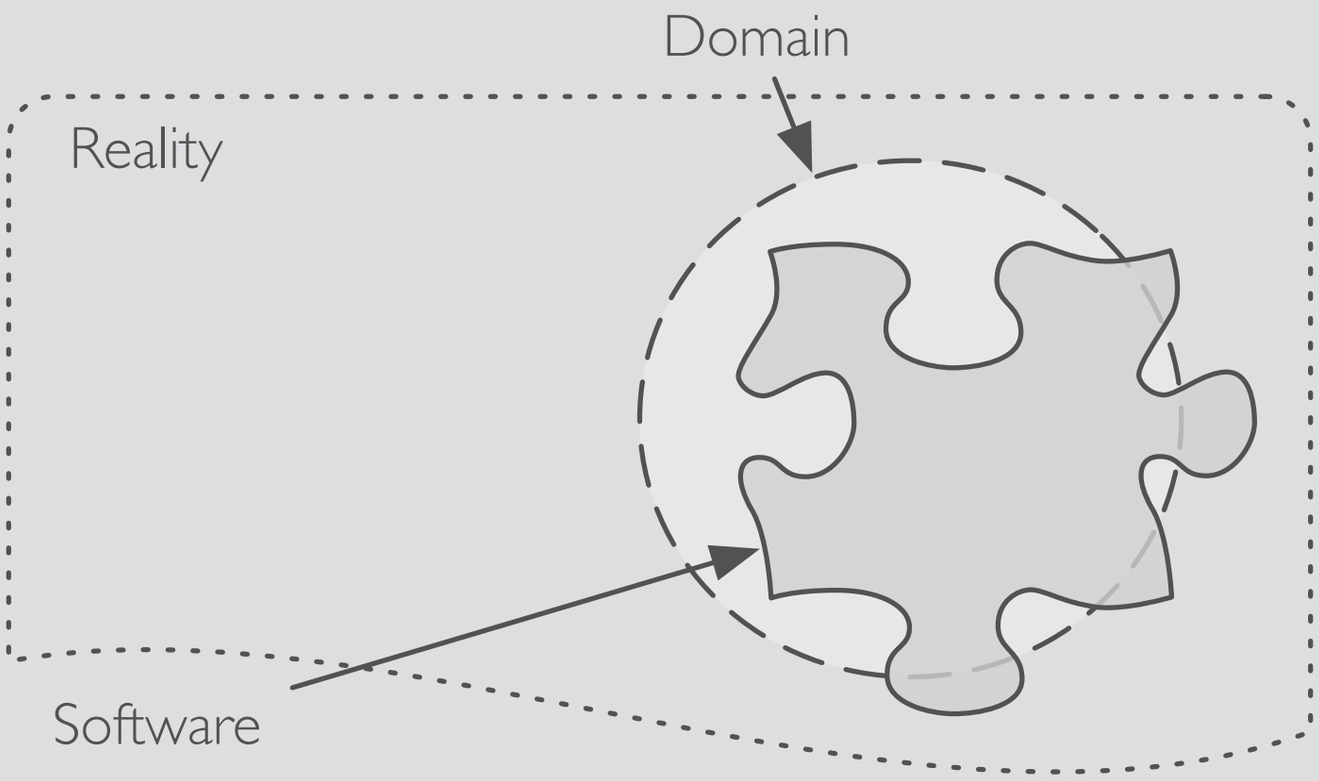
SUSPECT NUMBER ONE

What is the one thing you can always count on in software development?

- Independent of the tools…
- Independent of the problem domain…
- Independent of the task at hand…
- Independent of where you work…

CHANGE

THE STRUGGLE



No matter how well you design an application, over time an application must grow and change or it will die.

TYPES OF CHANGE

Requirements and Priorities:

- We learn from the solution: our true needs and how to communicate them better;
- Business environment and conditions change;
- Business processes are re-engineered…

TYPES OF CHANGE

Technology and Tools:

- We often learn new things on the fly,
- Actual features may vary from expectations,
- Combinations create compatibility issues,
- New versions are released...

TYPES OF CHANGE

People:

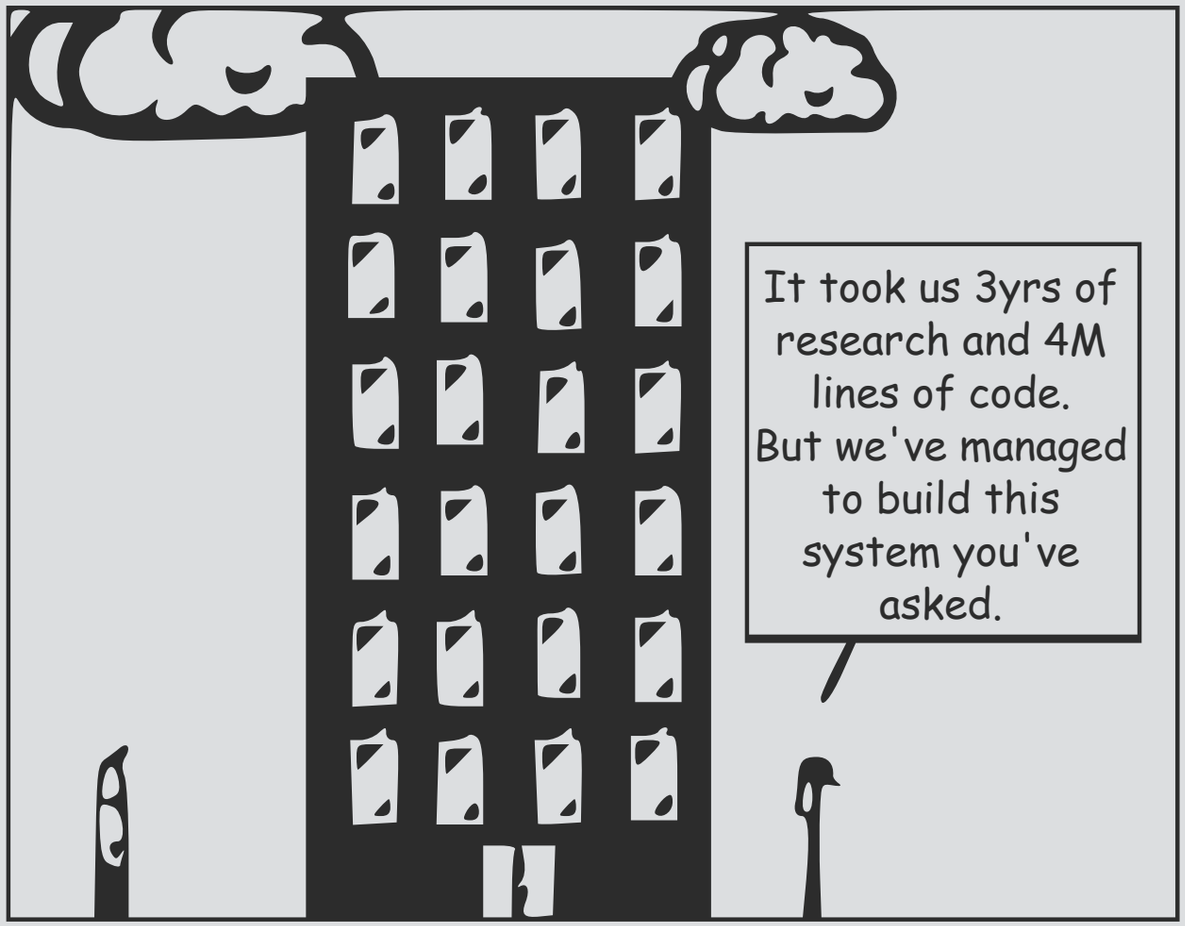
- Teams change over time,
- Team interactions may become complex,
- Individual behaviour can be unpredictable…

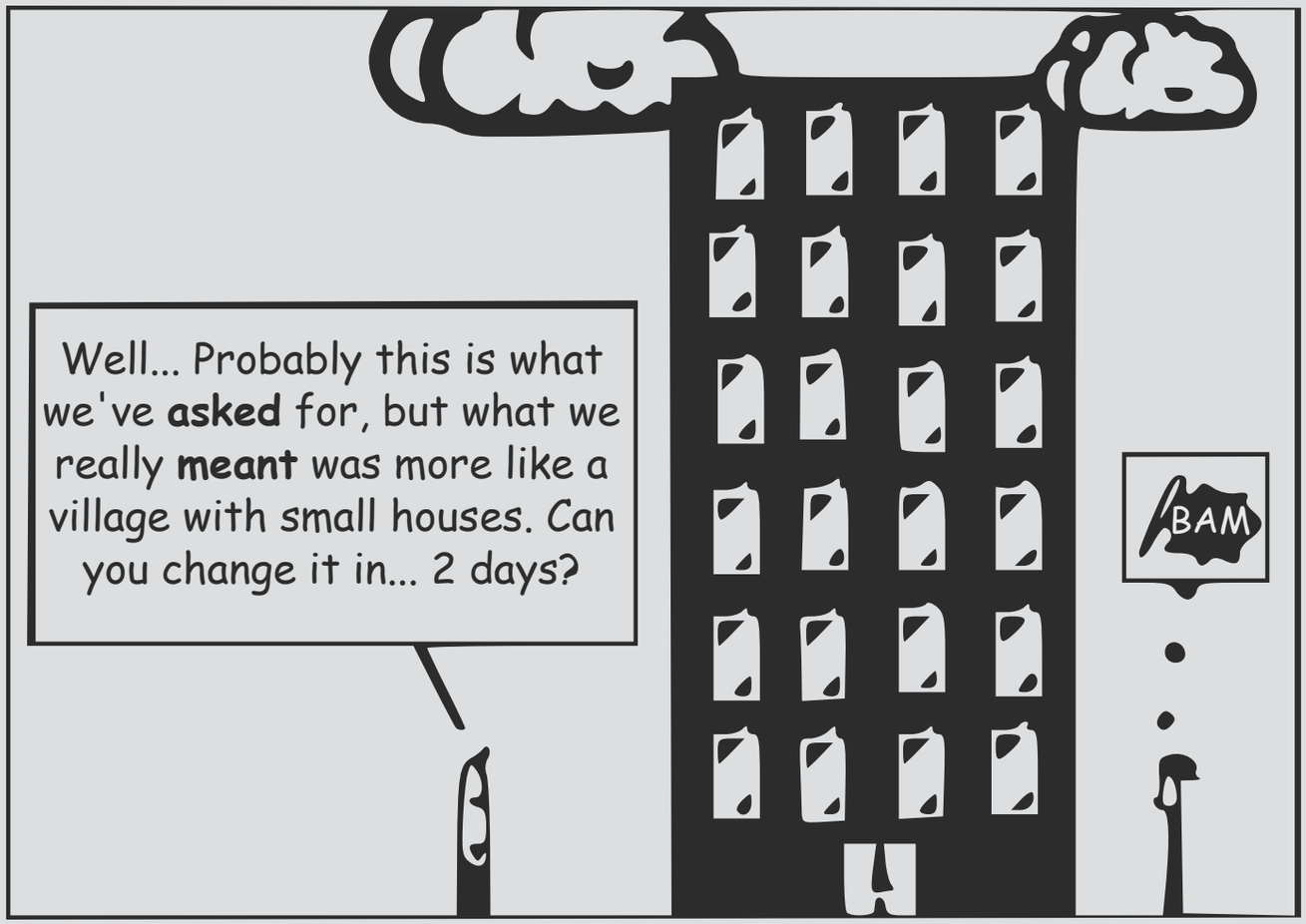
TYPES OF CHANGE

Project Complexity:

- Too much dependencies,
- Solutions need recursive feedback and validation,
- Difficulty to predict activities and dependencies.

TOO MUCH, TOO OFTEN





AGILE

> (1) marked by ready ability to move with quick easy grace; (2) having a quick resourceful and adaptable character: *an agile mind.* — *Merriam Webster*

For the purposes of this talk, it's the **ability to respond to change**.

HOW TO TACKLE CHANGE

WHY PROCESSES?

The challenge is to help achieve

- **High quality** of the developed product
- **High productivity** of development
- **Good predictability** of process results

WHY PROCESSES?

Define who? what? when? why? how?

- Roles
- Artefacts
- Activities
- Techniques
- Practices
- Tools

WHY PROCESSES?

Means to an end

- Suggest practices to help **improve team capabilities**
- Introduce formalities to **improve team discipline**
- Force documenting to **improve team communication and knowledge**

WHY PROCESSES?

Which practices, formalities and documentation?

- There is **no silver bullet™**!
- It **depends** on the project
- Balance them to your **needs**...

HEAVYWEIGHT

- Very preventive: try to **avoid** expensive situations instead of **optimising** them, even if avoiding such situations is more expensive than the original problem

- Requirements must be **exhaustively analysed**

- Search and **removal of errors before** appearing in the code

- Code is not considered very important, but only a **translation from specifications**

- Main problems: **reduced feedback** and **over-engineering**...

AGILE PROCESSES

- **Respond to change** and leverage learning
- Deliver the highest **business value** (ROI)
- Decrease time-to-delivery
- Increase productivity and efficiency
- Produce better **quality** solutions
- Create a more fulfilling development **culture**

AGILE VS HEAVYWEIGHT

Adopt the most simple process capable of achieving project's **success** and balance between discipline and agility. Five factors must be analysed before decision:

- **Criticality**, or errors impact
- **Dimension**, regarding number of team elements
- **Culture**, likelihood of achieving success in chaotic or ordered contexts
- **Dinamism**, or the frequency of changing requirements
- Team **technical capabilities**.

MANAGING AGILE

- Iterative and incremental
- Parallel and concurrent, not phased
- Planned around deliverables, not activities
- Dynamic project balancing via scope adjustments
- Heavy emphasis on collaboration
- Management by facilitation

ITERATIVE & INCREMENTAL

Iterative

- Repeatedly executing **nested process cycles**
- Iterations provide **synchronising points**
- Iterations provide **feedback points**

Incremental

- System is built in **progressive stages**
- Iterations add features and **refinements**
- Increments are **working systems**

PARALLEL AND CONCURRENT

Phased Approach

- Gathers **similar activity types together**
- Preference towards **serial completion**
- Ultimate in phased approach is **waterfall**

Concurrent and Parallel

- Activities occur **opportunistically**
- Activities of all types happening **simultaneously**
- **Partial completion** considered the norm

PREDICTIVE VS AGILE

Predictive Planning

- Creation of **comprehensive** activity-based plans
- Execution of defined activities to follow **plan**
- Management by controlling activities to **conform to plan**

Agile Planning

- Creation of **prioritised** set of deliverables
- **Opportunistic** execution of activities to create deliverables
- Management via **feedback and adaptation**

PROJECT BALANCE

- **Sustainable** resource management
  - **Stable** teams
  - **Steady** pace
    - Favor high ROI tools and technology
- Fixed **time management**: time-boxed cycles
- Adaptive **scope management**: feedback-based adjustments

HEROIC VS COLLABORATIVE

Heroic development emphasises **individuals**

- Activities **assigned** to individuals
- Project results heavily dependent on **individual performance**
- Increases **keyhole risks**

Collaborative development emphasises **teams**

- Teams self-organize activities to meet **goals**
- Teams leverage **diverse skills**
- Teams mitigate keyhole risks

MANAGING BY FACILITATION

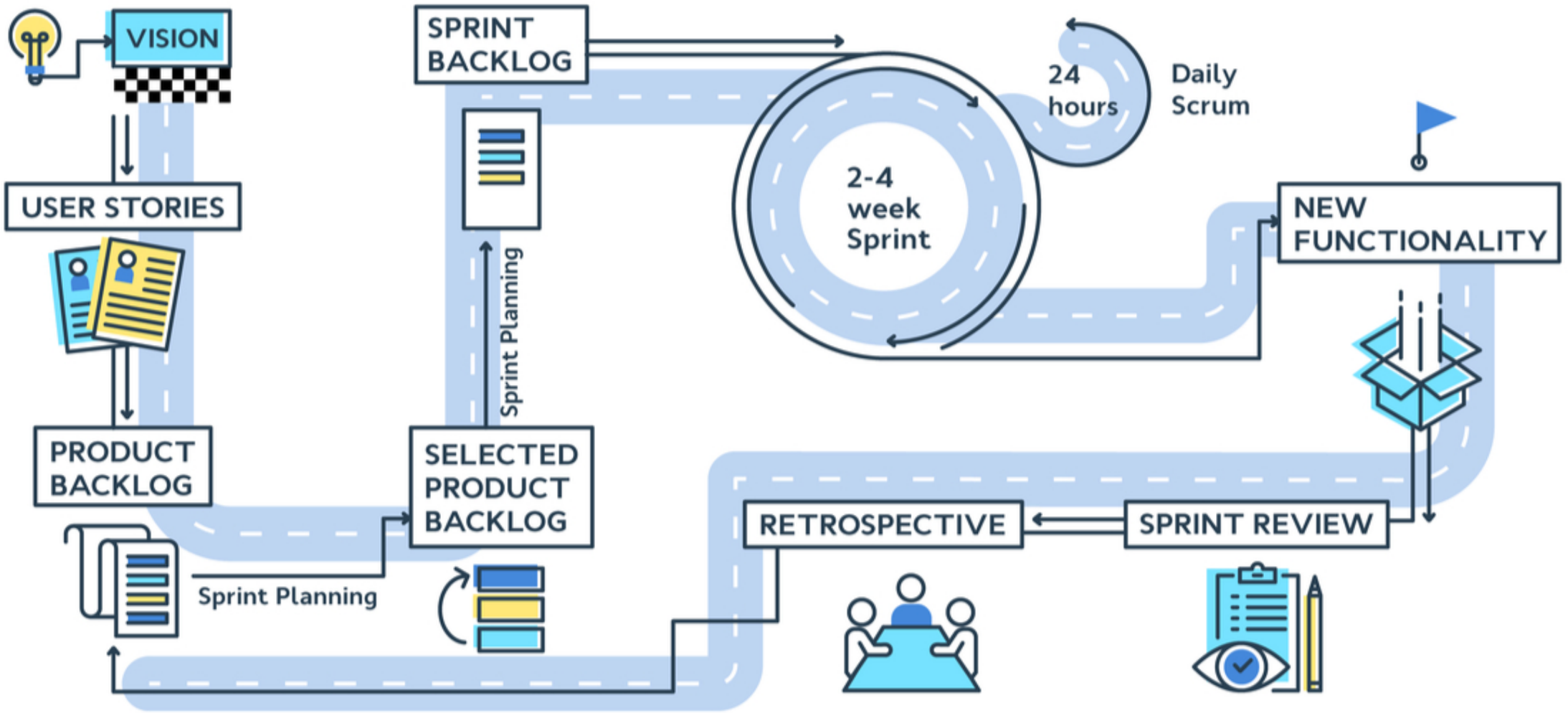
Command and Control Strategy

- Decisions made by **central authorities**
- Activities **delegated**
- Manager **controls** activities

Facilitation and Empowerment Strategy

- Decisions made by **those with the most info**
- Activities **accepted**
- Team **self-manages** and adapts
- Organisation ensures **supportive environment**

SCRUM PROCESS

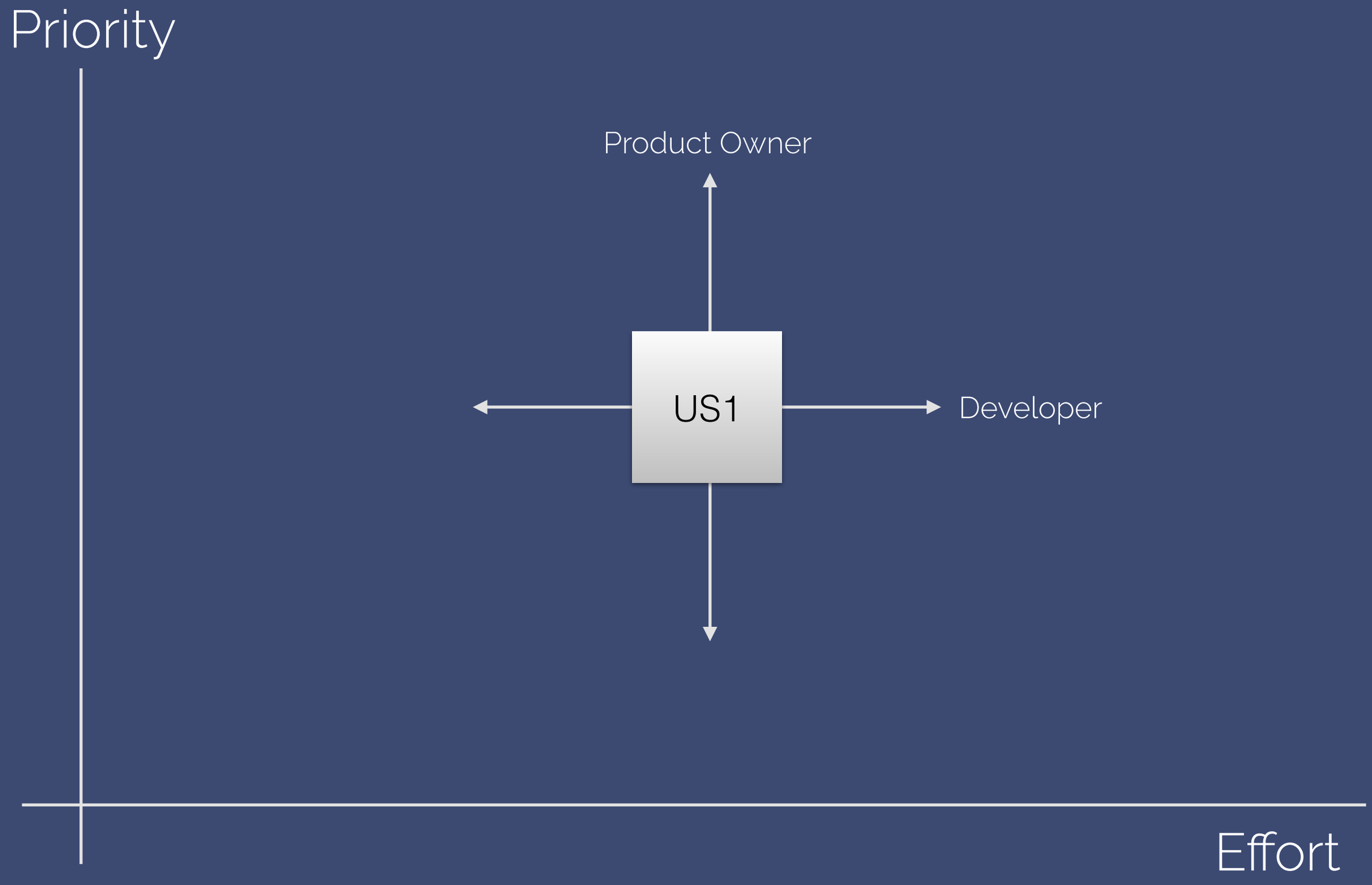


AGILE NEGOTIATION

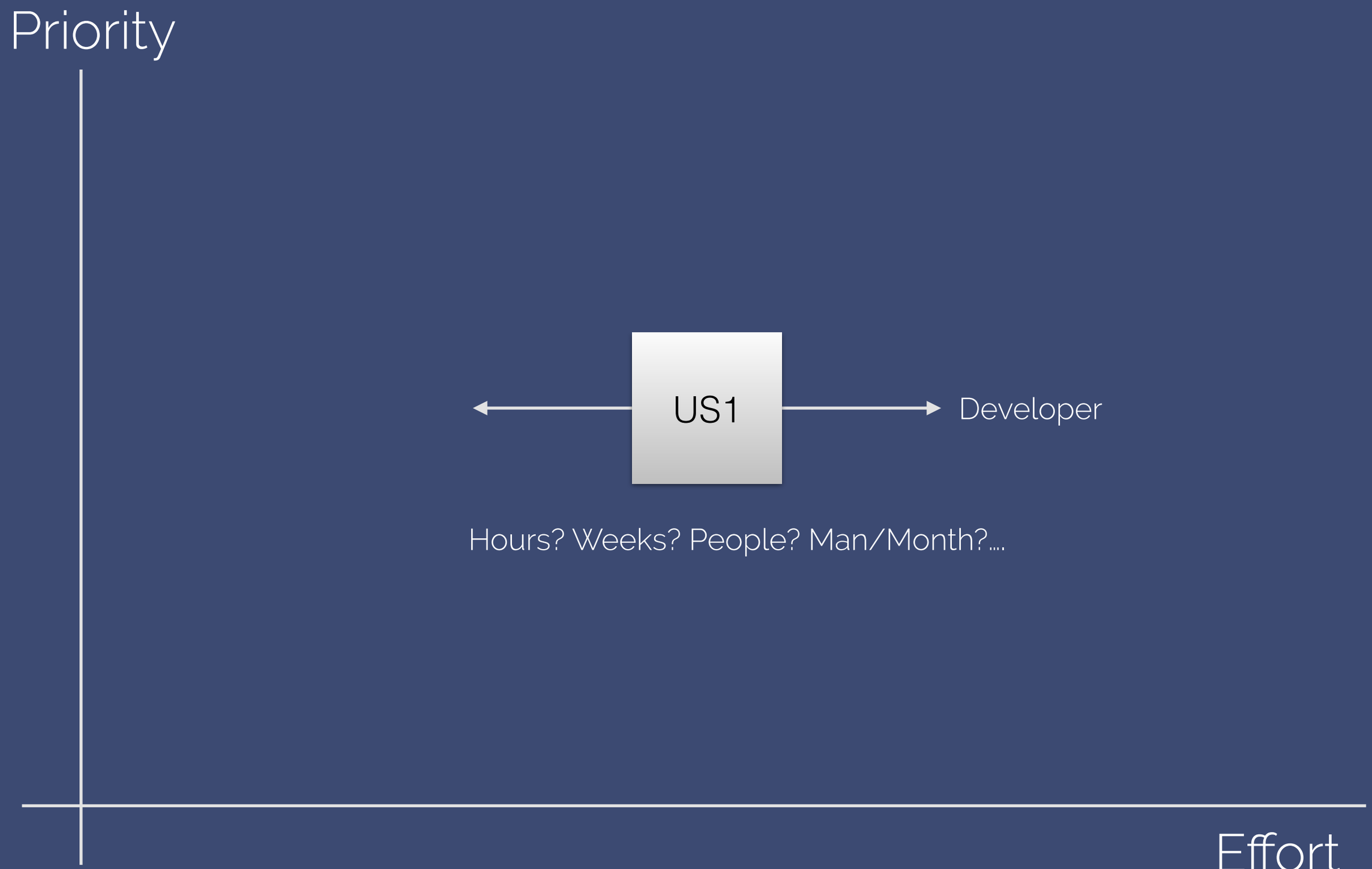
Priority

Effort

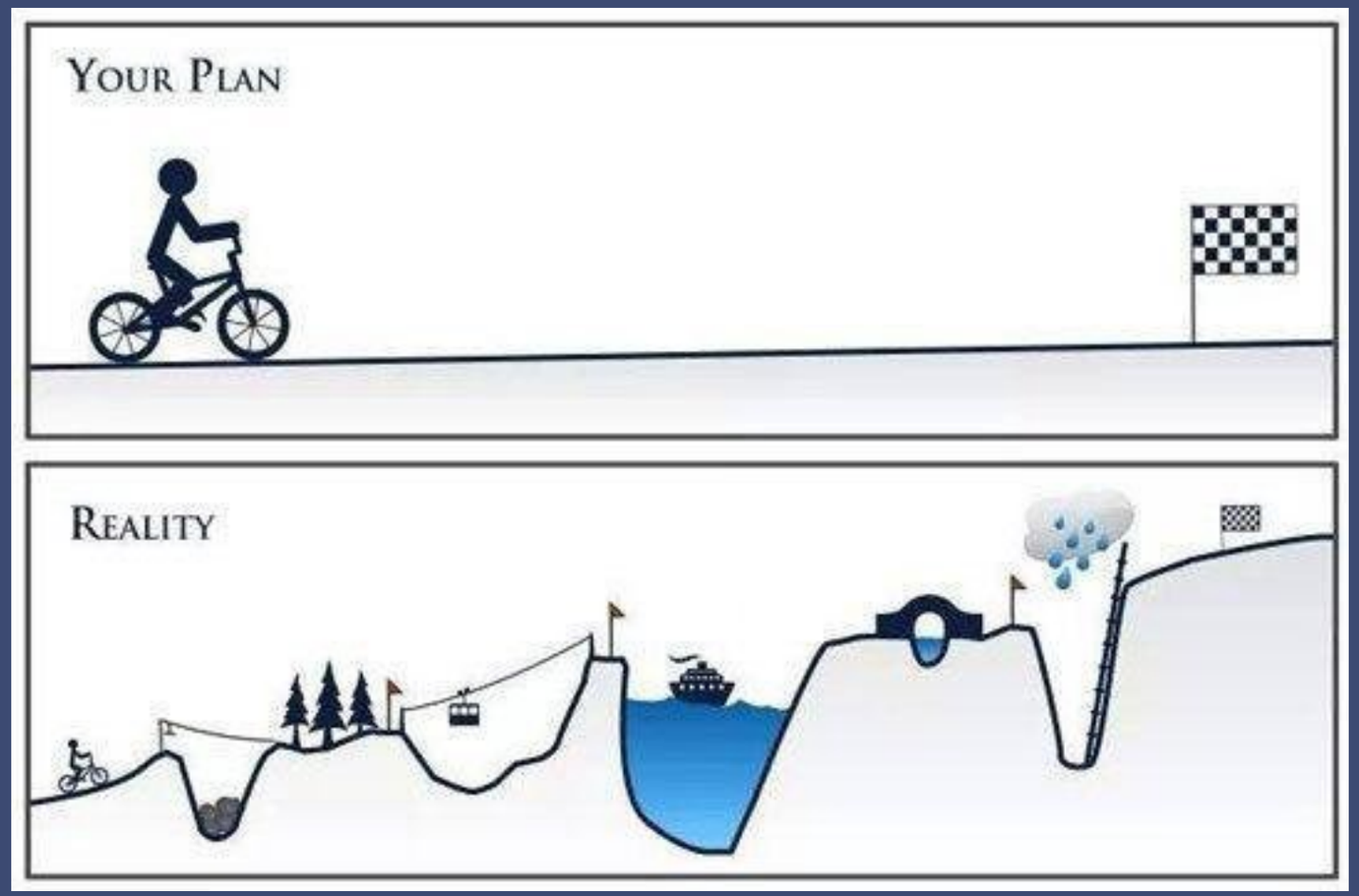
AGILE NEGOTIATION



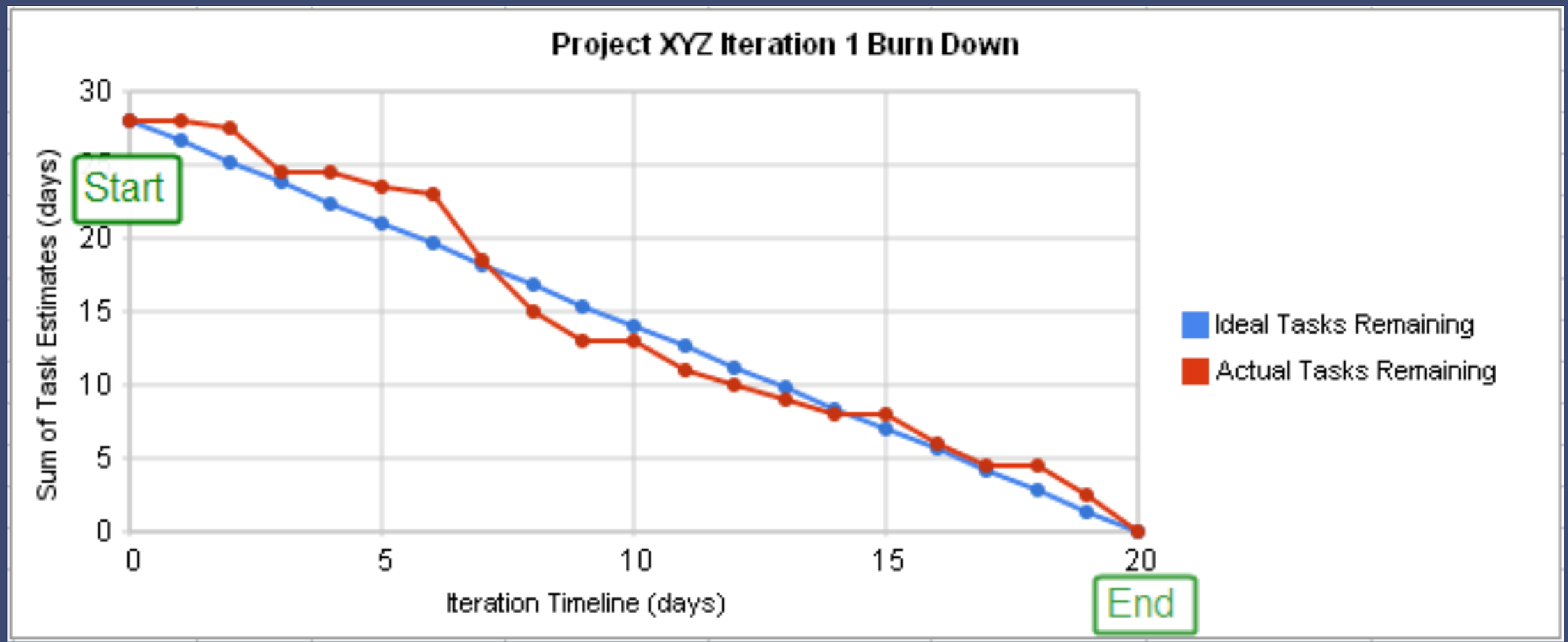
AGILE ESTIMATION



AGILE ESTIMATION



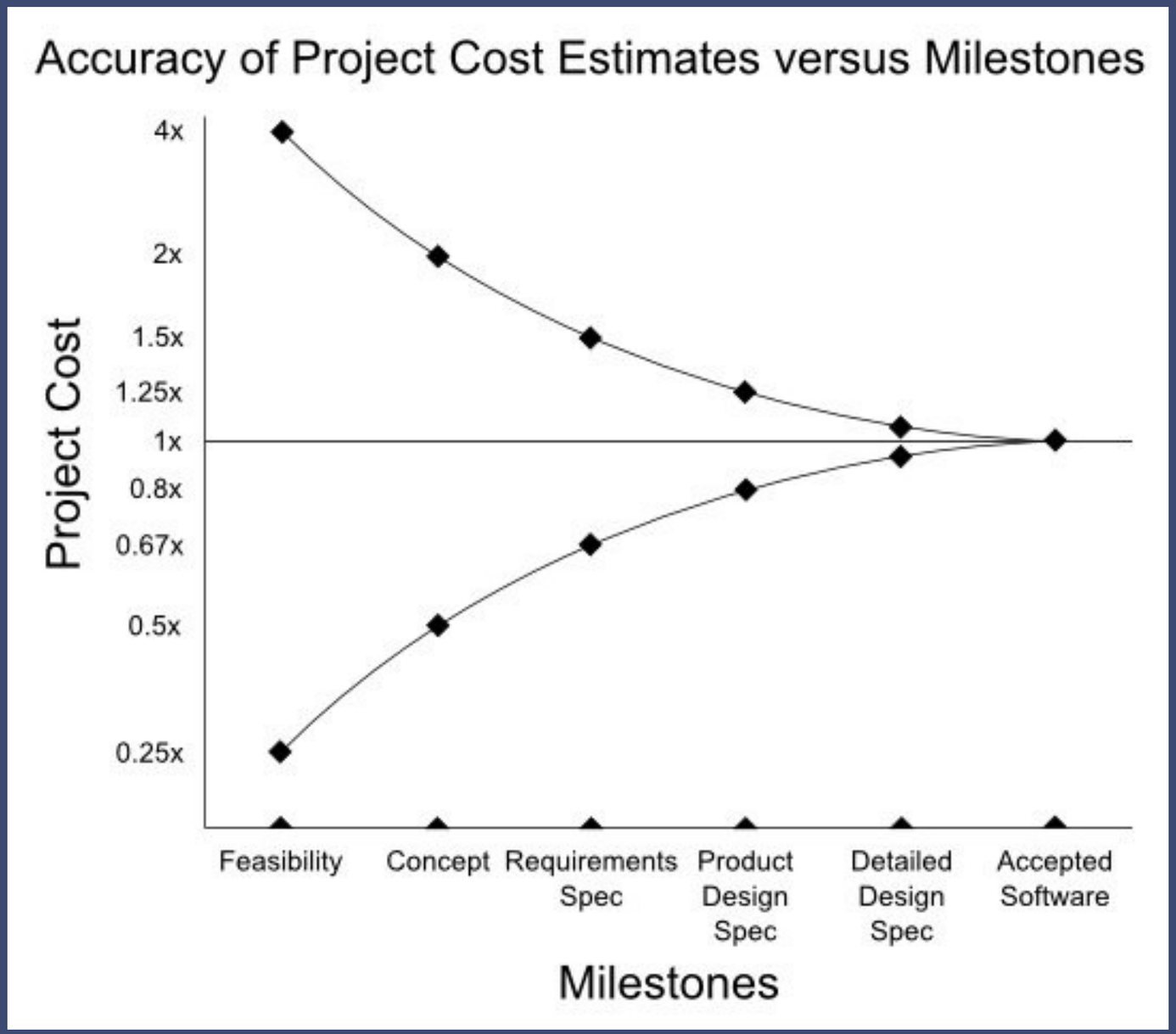
AGILE ESTIMATION



We are not interested in *per iteration* charts in LDSO

… we want to see the *whole project* burndown

AGILE ESTIMATION



LDSO

{hugo.sereno}@fe.up.pt