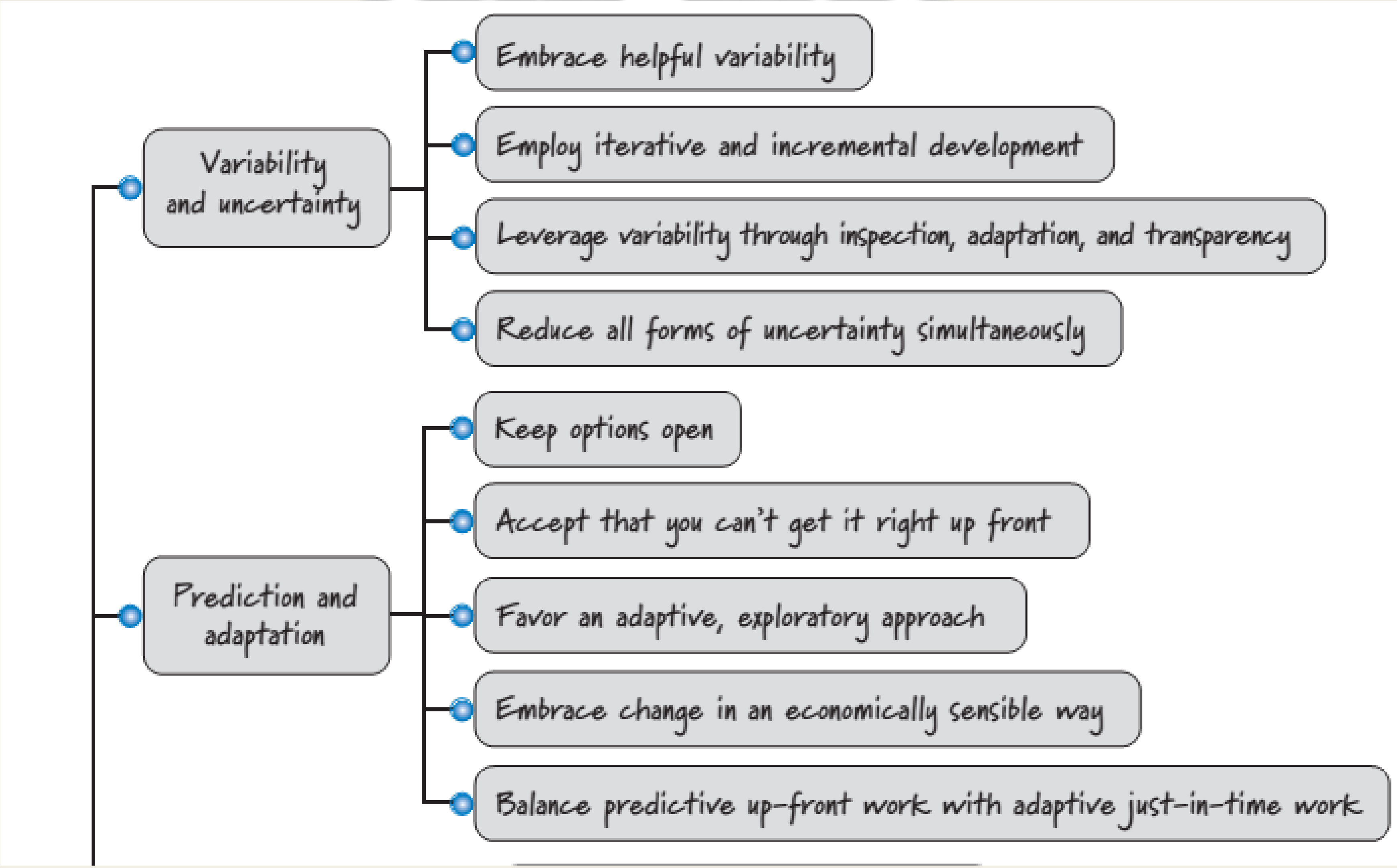
Agile Principles(III)

Dr. Elham Mahmoudzadeh
Isfahan University of Technology
mahmoudzadeh@iut.ac.ir

2022

Categorization of principles (Up)



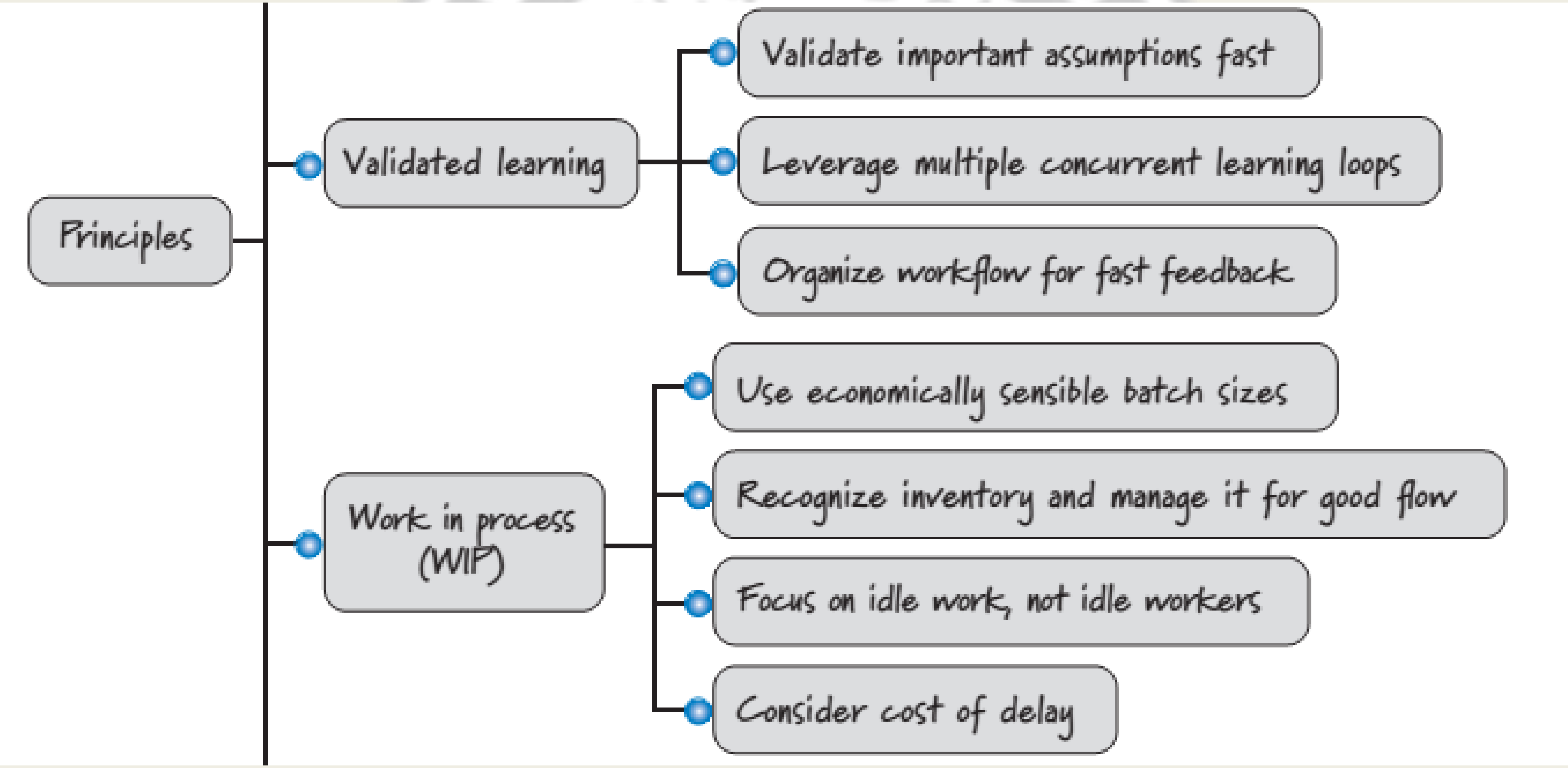
Variability and uncertainty

- Embrace helpful variability
- Employ iterative and incremental development
- Leverage variability through inspection, adaptation, and transparency
- Reduce all forms of uncertainty simultaneously

Prediction and adaptation

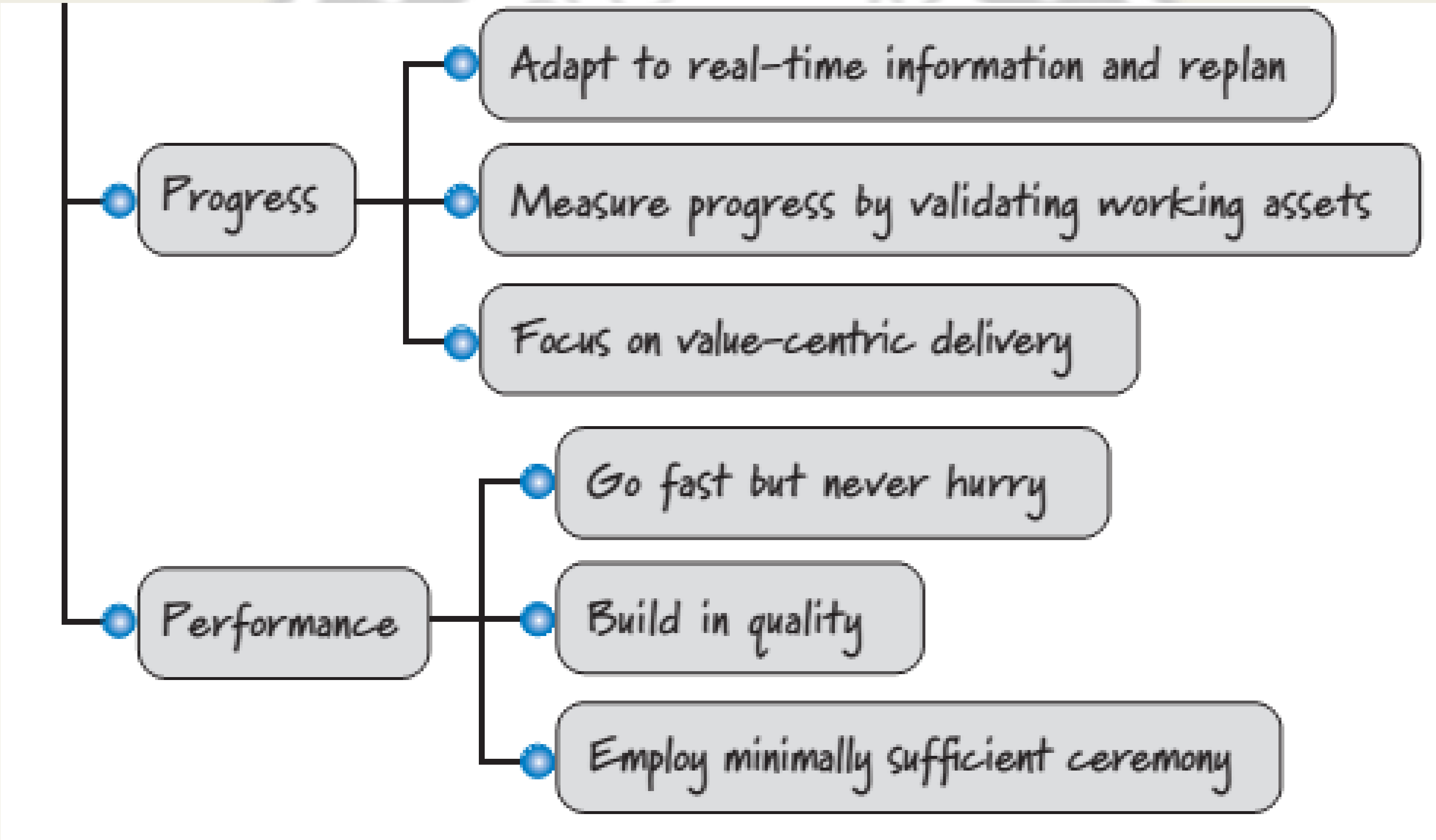
- Keep options open
- Accept that you can't get it right up front
- Favor an adaptive, exploratory approach
- Embrace change in an economically sensible way
- Balance predictive up-front work with adaptive just-in-time work

Categorization of principles (Middle)



- Principles
  - Validated learning
    - Validate important assumptions fast
    - Leverage multiple concurrent learning loops
    - Organize workflow for fast feedback
  - Work in process (WIP)
    - Use economically sensible batch sizes
    - Recognize inventory and manage it for good flow
    - Focus on idle work, not idle workers
    - Consider cost of delay

Categorization of principles (Bottom)


Progress
Adapt to real-time information and replan
Measure progress by validating working assets
Focus on value-centric delivery
Performance
Go fast but never hurry
Build in quality
Employ minimally sufficient ceremony

Prediction and Adaptation

- In Scrum, we are constantly balancing the desire for prediction with the need for adaptation.
- Five related principles
    1. *Keep options open.*
    2. *Accept that you can't get it right up front.*
    3. *Favor an adaptive, exploratory approach.*
    4. *Embrace change in an economically sensible way.*
    5. *Balance predictive up-front work with adaptive just-in-time work.*

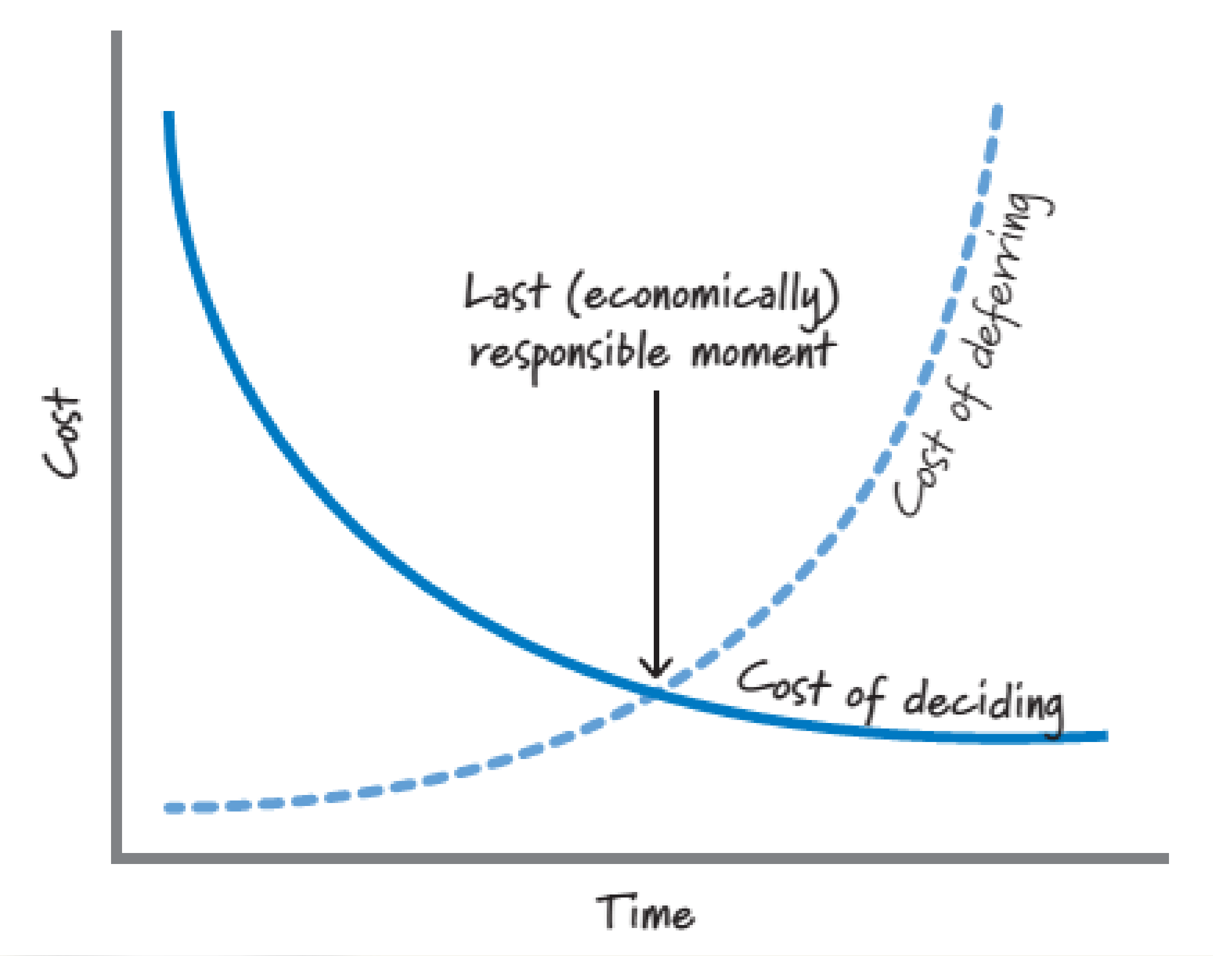
Keep Options Open (I)

- Plan-driven, sequential development requires that important decisions in areas like requirements or design be made, reviewed, and approved within their respective phases.
- Furthermore, these decisions must be made before we can transit to the next phase, even if those decisions are based on limited knowledge.

Keep Options Open (II)

- Scrum contends that we should never make a premature decision just because a generic process would dictate that now is the appointed time to make one.
- In Scrum, we favor a strategy of keeping our options open.
- Often this principle is referred to as the **last responsible moment (LRM)**, meaning that we delay commitment and do not make important and irreversible decisions until the last responsible moment. <u>LRM is when the cost of not making a decision becomes greater than the cost of making a decision</u>. At that moment, we make the decision.

Keep Options Open (III)



Keep Options Open (IV)

- On the first day of a product development effort, we have the least information about what we are doing.

- On each subsequent day of the development effort, we learn a little more.

- Most of us would prefer to wait until we have more information so that we can make a more informed decision.

Keep Options Open (V)

- When dealing with important or irreversible decisions, if we decide too early and are wrong, we will be on the exponential part of the cost-of-deciding curve.

- As we acquire a better understanding regarding the decision, the cost of deciding declines (the likelihood of making a bad decision declines because of increasing market or technical certainty). That’s why we should wait until we have better information before committing to a decision.

Accept That You Can’t Get It Right Up Front(I)

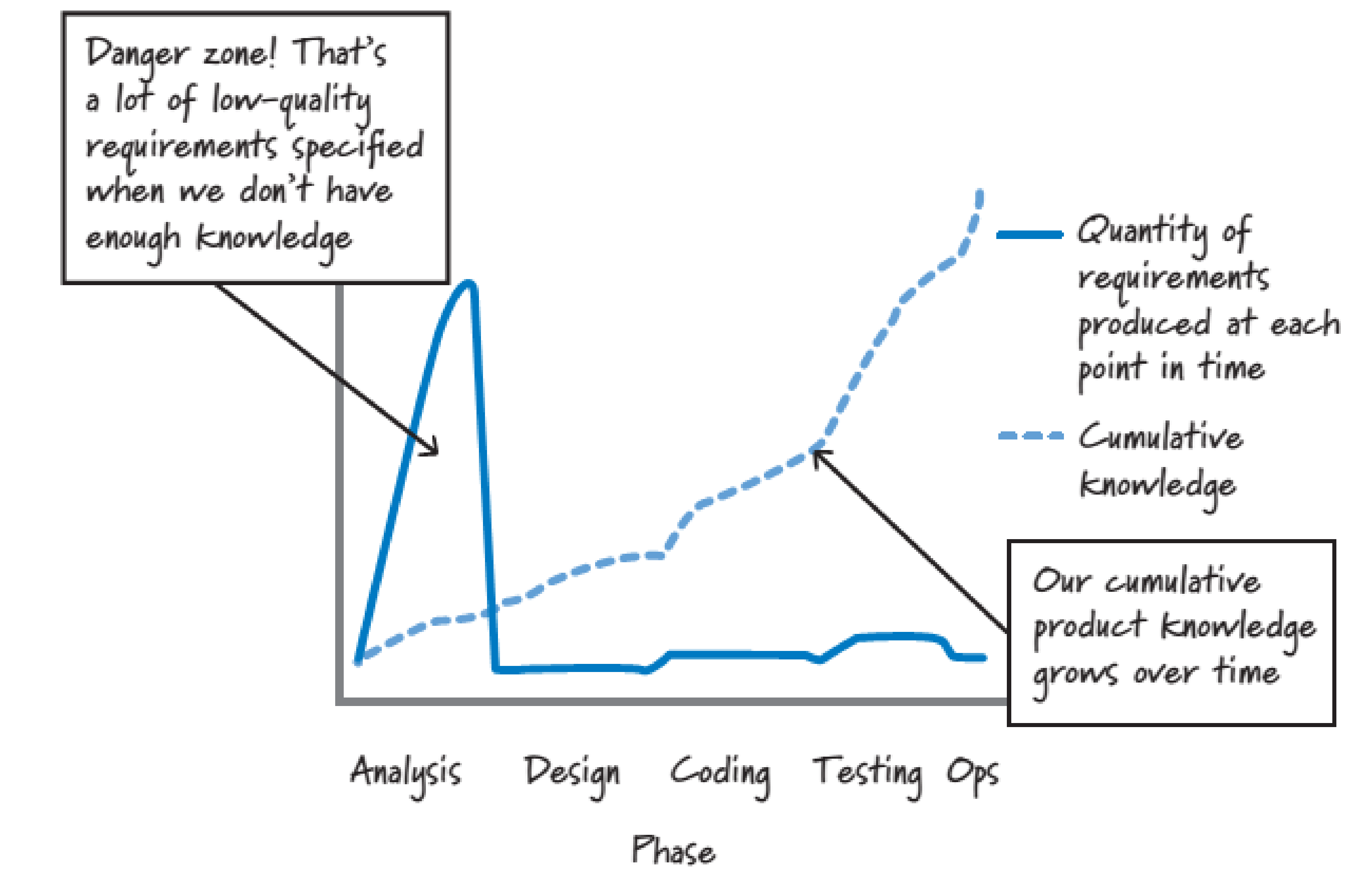
- Plan-driven processes not only mandate full requirements and a complete plan; they also assume that we can “get it right” up front.
- The reality is that it is very unlikely that we can get all of the requirements, or the detailed plans based on those requirements, correct up front.
- What’s worse is that when the requirements do change, we have to modify the baseline requirements and plans to match the current reality.

Accept That You Can’t Get It Right Up Front(II)

- In Scrum, we acknowledge that we can’t get all of the requirements or the plans right up front.

- In fact, we believe that trying to do so could be dangerous because we are likely missing important knowledge, leading to the creation of a large quantity of low-quality requirement.

Plan-driven requirements acquisition relative to product knowledge



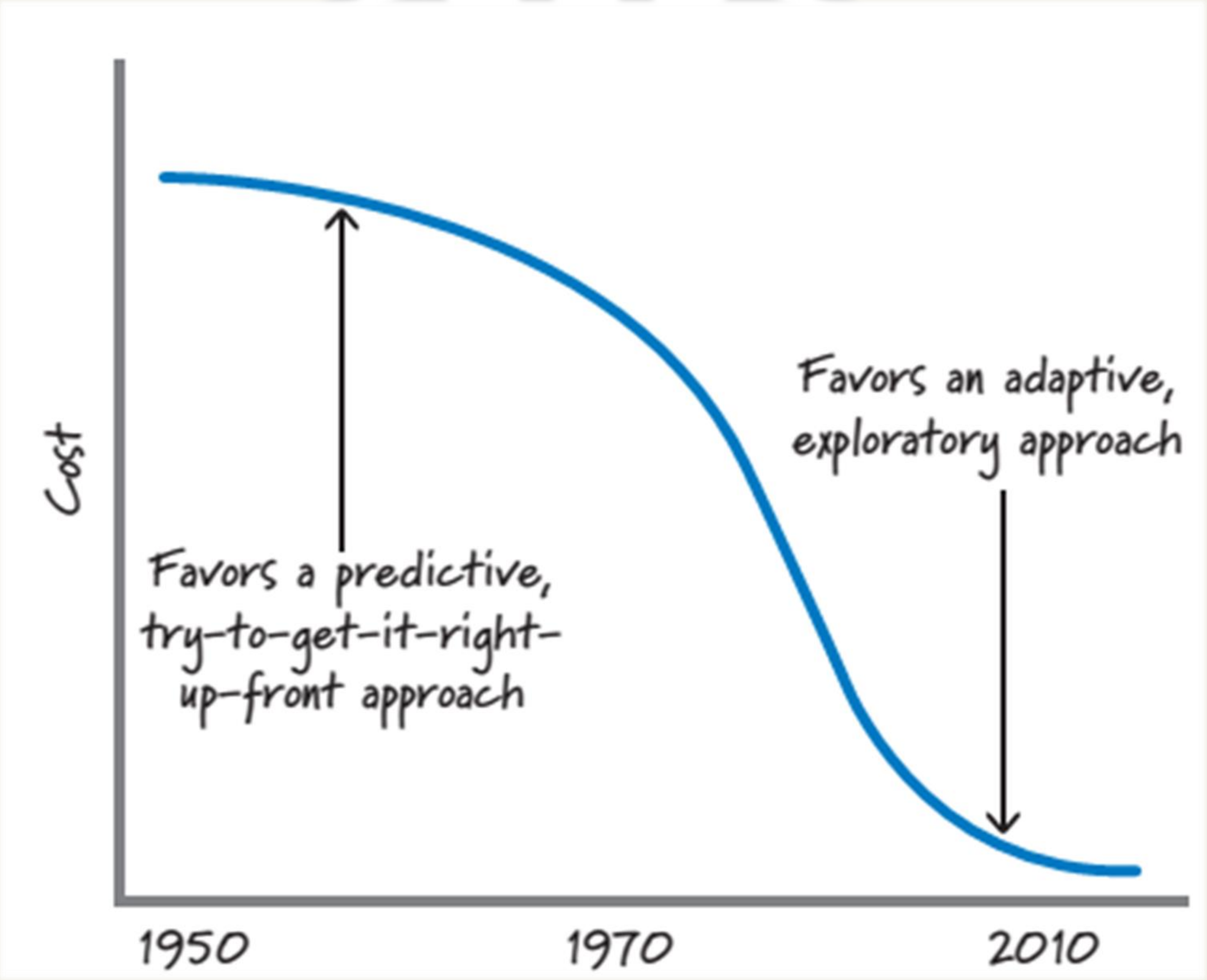
Accept That You Can’t Get It Right Up Front(IV)

- With Scrum, we still produce some requirements and plans up front, but just sufficiently, and with the assumption that we will fill in the details of those requirements and plans as we learn more about the product we are building.
- After all, even if we think we’re 100% certain about what to build and how to organize up front the work to build it, we will learn where we are wrong as soon as we subject our early incremental deliverables to the environment in which they must exist. At that point all of the inconvenient realities of what is really needed will drive us to make changes.

Favor an Adaptive, Exploratory Approach (I)

- **Exploration** refers to times when we choose to gain knowledge by doing some activity, such as building a prototype, performing a study, or conducting an experiment. In other words, when faced with uncertainty, we buy information by exploring.
- Plan-driven, sequential processes focus on using (or exploiting) what is currently known and predicting what isn’t known.
- Scrum favors a more adaptive, trial-and error approach based on appropriate use of exploration.
- Our tools and technologies significantly influence the cost of exploration.

Historical cost of exploration



Favor an Adaptive, Exploratory Approach(III)

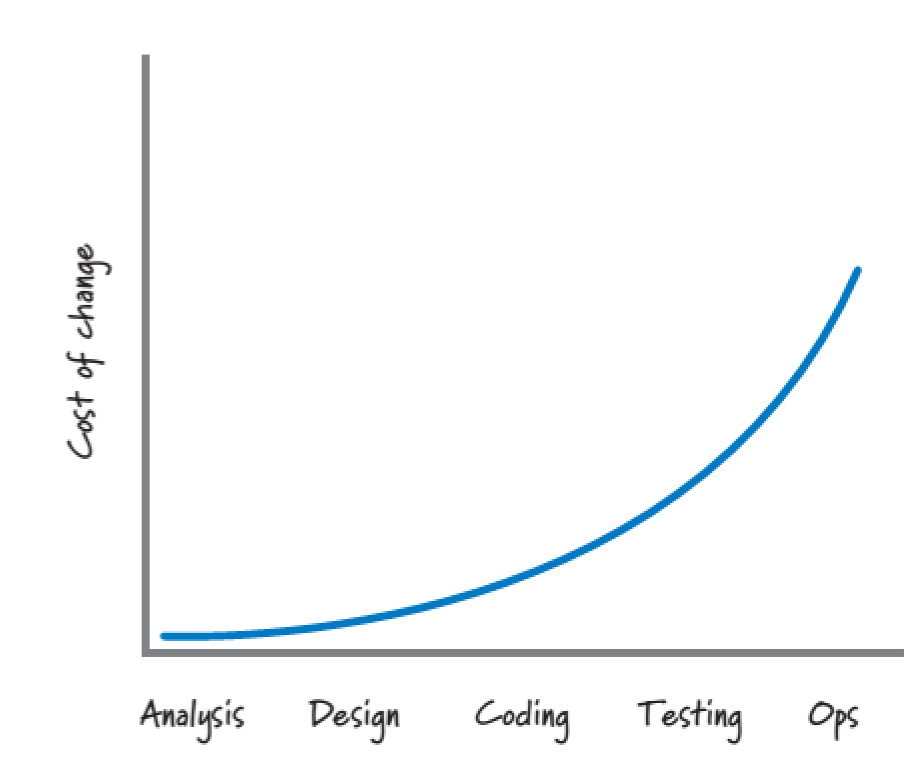
- Tools and technologies have gotten better and the cost of exploring has come way down.
- In fact, nowadays, it’s often cheaper to adapt to user feedback based on building something fast than it is to invest in trying to get everything right up front.

Favor an Adaptive, Exploratory Approach(IV)

- In Scrum, if we have enough knowledge to make an informed, reasonable step forward with our solution, we advance.
- When faced with uncertainty, rather than trying to predict it away, we use low-cost exploration to buy relevant information that we can then use to make an informed, reasonable step forward with our solution.
- The feedback from our action will help us determine if and when we need further exploration.

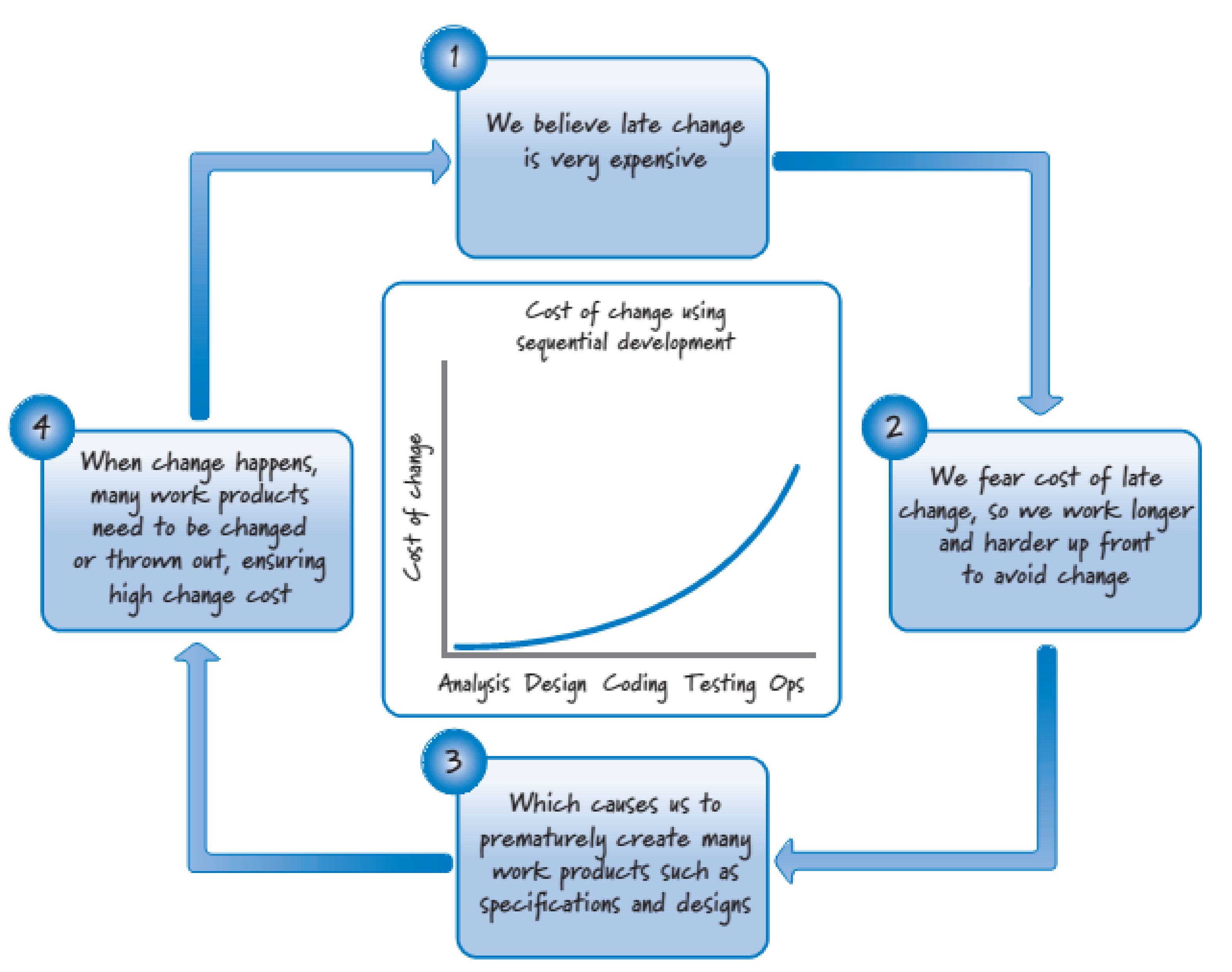
Embrace Change in an Economically Sensible Way(I)

- When using sequential development, change, as we have all learned, is substantially more expensive late than it is early on.
- To avoid late changes, sequential processes seek to carefully control and minimize any changing requirements or designs by improving the accuracy of the predictions about what the system needs to do or how it is supposed to do it.



Embrace Change in an Economically Sensible Way(II)

- Unfortunately, being excessively predictive in early-activity phases often has the opposite effect.
- First, the desire to eliminate expensive change forces us to overinvest in each phase—doing more work than is necessary and practical.
- Second, we're forced to make decisions based on important assumptions early in the process, before we have validated these assumptions with feedback from our stakeholders based on our working assets. As a result, we produce a large inventory of work products based on these assumptions. Later, this inventory will likely have to be corrected or discarded as we validate (or invalidate) our assumptions, or change happens.

1
We believe late change is very expensive
2
We fear cost of late change, so we work longer and harder up front to avoid change
3
Which causes us to prematurely create many work products such as specifications and designs
4
When change happens, many work products need to be changed or thrown out, ensuring high change cost
Cost of change using sequential development
Cost of change
Analysis
Design
Coding
Testing
Ops

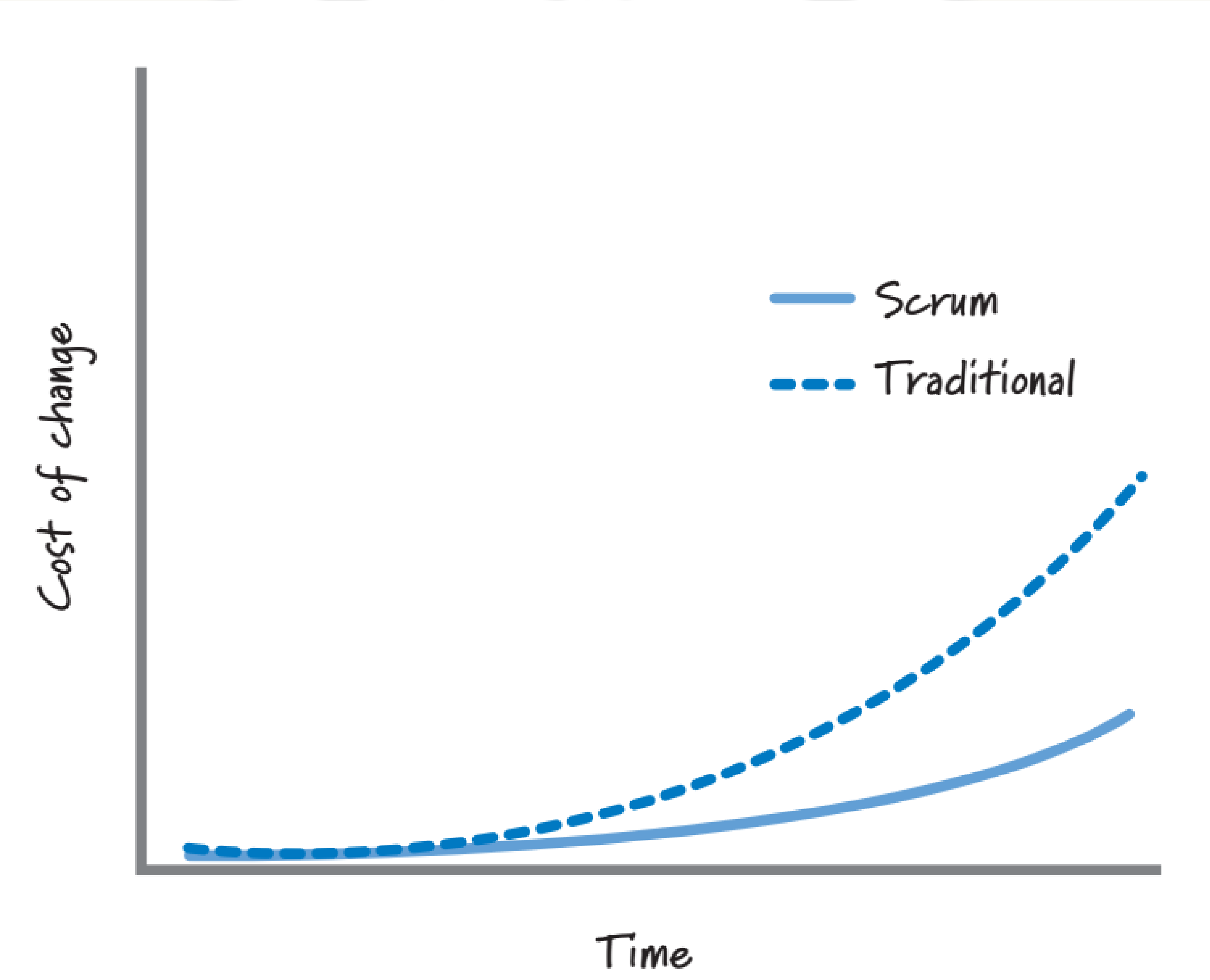
Embrace Change in an Economically Sensible Way(IV)

- In Scrum, we assume that change is the norm.
- We believe that we can’t predict away the inherent uncertainty that exists during product development by working longer and harder up front.
- Thus, we must be prepared to embrace change. And when that change occurs, we want the economics to be more appealing than with traditional development, even when the change happens later in the product development effort.

Embrace Change in an Economically Sensible Way(V)

- Our goal, t for as long
  as possi e even late
  change.

- We can of work in
  process ange when
  using Sc sequential
  projects.



Embrace Change in an Economically Sensible Way(VI)

- Regardless of which product development approach we use, we want the following relationship to be true.
  - *A small change in requirements should yield a proportionally small change in implementation and therefore in cost (obviously we would expect a larger change to cost more).*
- Another desirable property that we want it to be true regardless of <u>when the change request is made</u>.

Embrace Change in an Economically Sensible Way(VII)

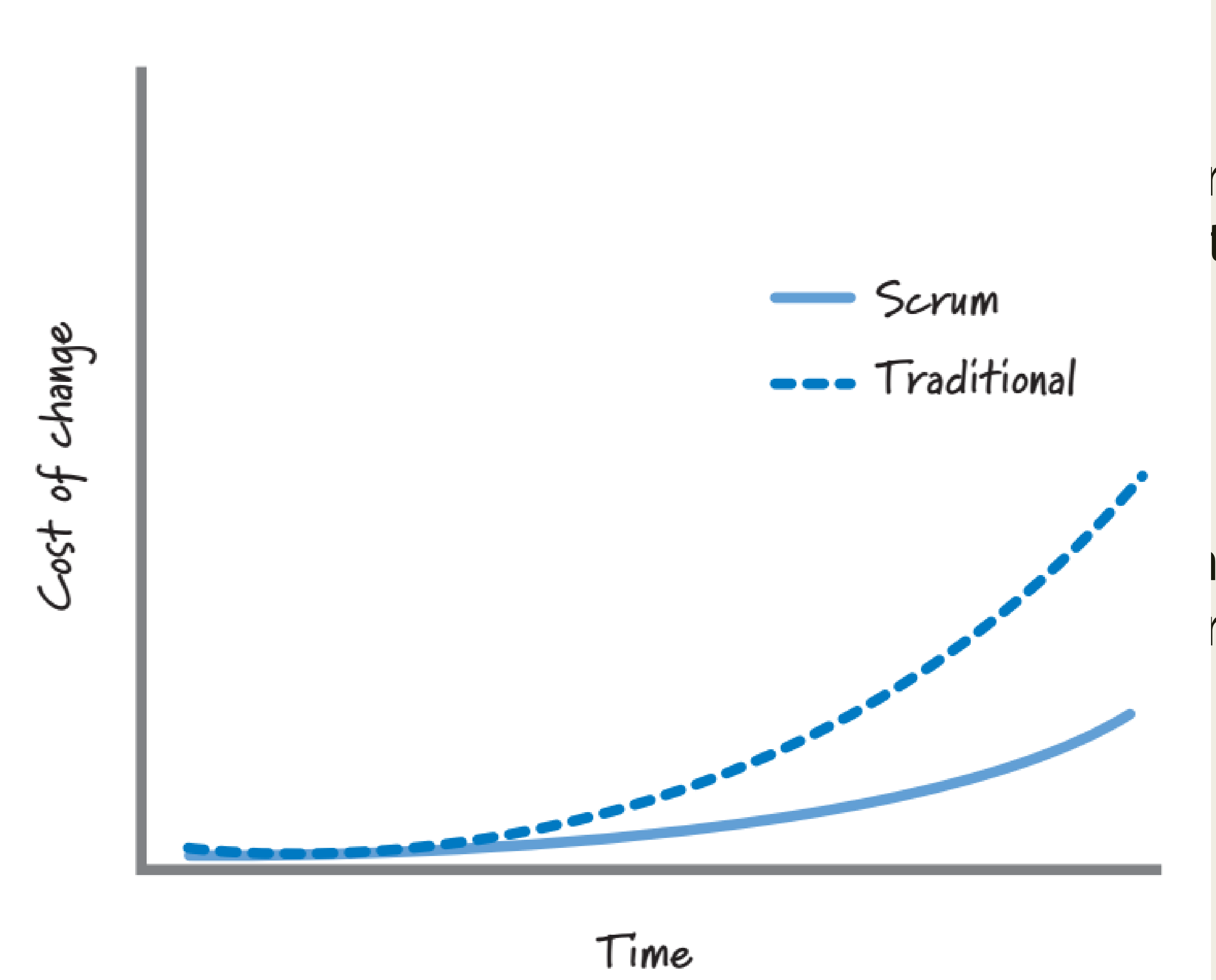
- With Scrum, we produce many work products (such as detailed requirements, designs, and test cases) in a just-in-time fashion, avoiding the creation of potentially unnecessary artifacts.

- As a result, when a change is made, there are typically far fewer artifacts or constraining decisions based on assumptions that might be discarded or reworked, thus keeping the cost more proportional to the size of the requested change.

Embrace Change in an Economically Sensible Way(VIII)

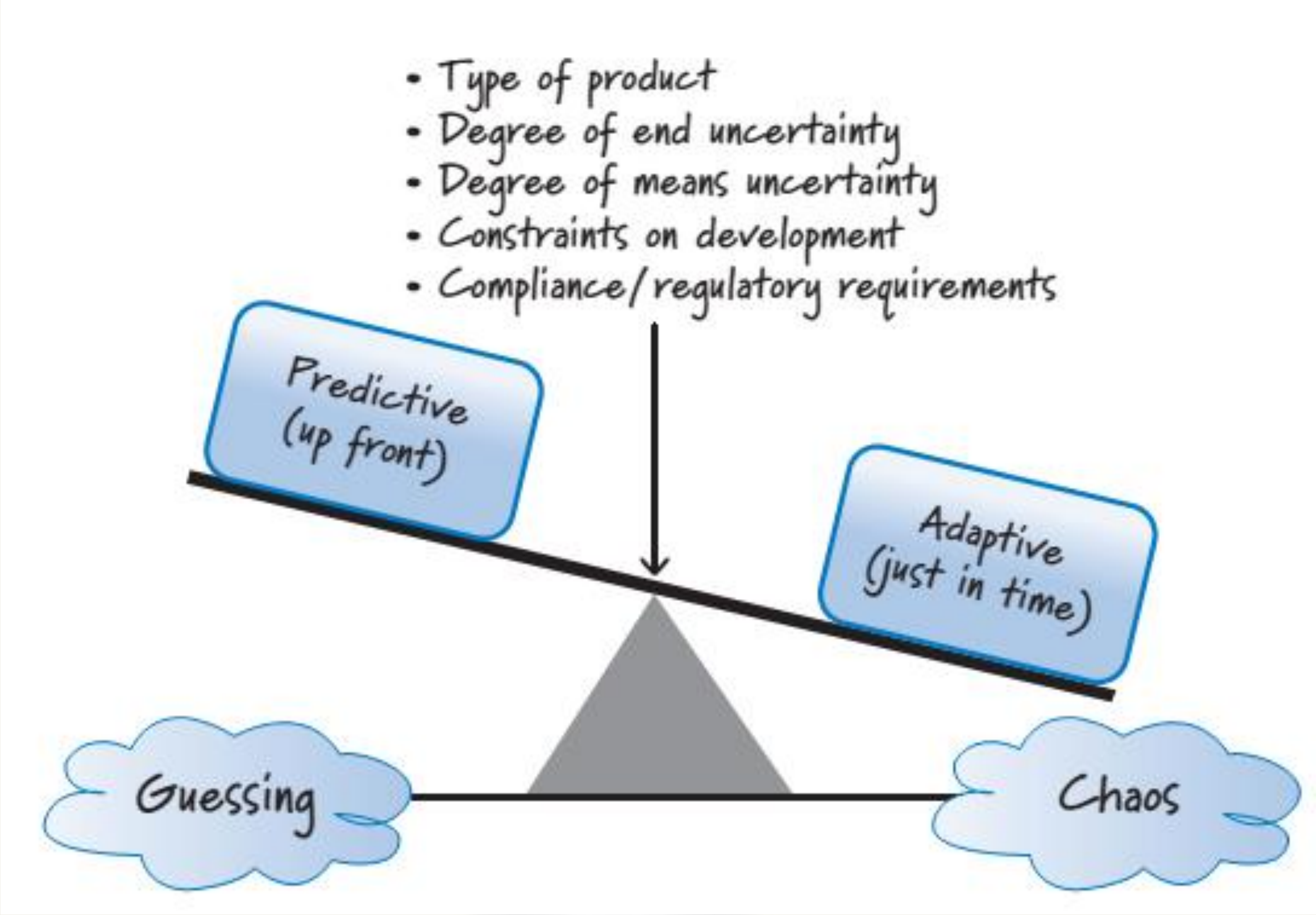
- Using se rtifacts and push for t the cost of a change causes the inflectior

- When de ien the cost of chang request, but this point



Balance Predictive Up-Front Work with Adaptive Just-in-Time Work(I)

- A fundamental belief of plan-driven development is that detailed up-front requirements and planning are critical and should be completed before moving on to later stages.
- In Scrum, we believe that up-front work should be helpful without being excessive. With Scrum, we acknowledge that it is not possible to get requirements and plans precisely right up front.
- Scrum is about finding balance—balance between predictive up-front work and adaptive just-in-time work.


• Type of product
• Degree of end uncertainty
• Degree of means uncertainty
• Constraints on development
• Compliance/regulatory requirements
Predictive (up front)
Adaptive (just in time)
Guessing
Chaos

Balance Predictive Up-Front Work with Adaptive Just-in-Time Work(III)

- When developing a product, the balance point should be set in an economically sensible way to maximize the amount of ongoing adaptation based on fast feedback and minimize the amount of up-front prediction, while still meeting compliance, regulatory, and/or corporate objectives.

- Exactly how that balance is achieved is driven in part by the type of product being built, the degree of uncertainty that exists in both what we want to build (end uncertainty) and how we want to build it (means uncertainty), and the constraints placed on the development.

Balance Predictive Up-Front Work with Adaptive Just-in-Time Work(IV)

- Being overly predictive would require us to make many assumptions in the presence of great uncertainty. Being overly adaptive could cause us to live in a state of constant change, making our work feel inefficient and chaotic.

- To rapidly develop innovative products we need to operate in a space where adaptability is counterbalanced by just enough prediction to keep us from sliding into chaos.
- The Scrum framework operates well at this balance point of order and chaos.

Reference

1- K. S. Rubin, “Essential Scrum, A Practical guide to the most popular agile process,” 2013.