Agile Software Development

Lecturer: Raman Ramsin

Lecture 4

Scrum Applicability
Scrum Benefits

- Delighted customers
- Improved return on investment
- Reduced costs
- Fast results
- Confidence to succeed in a complex world
- More joy
Applicability of Scrum

- Scrum is not the proper solution in all problem situations.
- We will discuss Scrum’s applicability based on the categories of problem situations proposed by the Cynefin Framework.
  - Cynefin, pronounced ‘ku-nev-in’, is a Welsh word for habitat:
    - It signifies the factors in our environment/experience that influence us in incomprehensible ways.
  - The Cynefin Framework is a sense-making framework that helps us understand the situation in which we have to operate, and decide on a situation-appropriate approach.
    - Defines and compares the characteristics of five different domains: Simple (Obvious), Complicated, Chaotic, Complex, and Disorder.
Cynefin Framework

**Complex**
- Probe, Sense, Respond
- Explore to learn about problem, then inspect, and then adapt
- Requires creative/innovative approaches
- Create safe-fail environment for experimentation to discover patterns
- Increase levels of interaction/communication
- Domain of emergence
- We’ll know in hindsight
- More unpredictable than predictable

**Complicated**
- Sense, Analyze, Respond
- Assess the situation, investigate several options, base response on good practice
- Use experts to gain insight
- Use metrics to gain control
- Domain of good practices
- Multiple right answers
- Cause and effect are discoverable but not immediately apparent
- More predictable than unpredictable

**Chaotic**
- Act, Sense, Respond
- Act immediately, then inspect to see if situation has stabilized, then adapt to try to migrate context to complex domain
- Many decisions to make; no time to think
- Immediate action to reestablish order
- Look for what works instead of right answers
- Domain of the novel
- No one knows
- No clear cause and effect

**Simple**
- Sense, Categorize, Respond
- Assess situation facts, categorize them, base response on established practice
- Domain of best practices
- Stable domain (not likely to change)
- Clear cause-and-effect relationships are evident to everyone
- A correct answer exists
- Fact-based management

[Rubin 2012]
Cynefin: Complex Domain

- In complex problems, things are more unpredictable than they are predictable.
- If there is a right answer, we will know it only with hindsight; this is the domain of emergence.
- We need to **probe** to learn about the problem, then **sense** and **respond** based on our learning.
- Working in complex domains requires creative and innovative approaches. Routine solutions simply don’t apply.
- We need to create a safe-fail environment for experimentation so that we can discover important information.
- In this environment high levels of interaction and communication are essential.
- Innovative new-product development falls into this category, as does enhancing existing products with innovative new features.
- **Scrum** is particularly well suited for operating in a complex domain, where our ability to **explore** (probe), **inspect** (sense), and **adapt** (respond) is critical.
Cynefin: Complicated Domain

- In complicated problems, things are more predictable than unpredictable.
- Complicated problems are the domain of good practices dominated by experts.
- There might be multiple right answers, but expert diagnosis is required to figure them out.
- The approach is to **sense** what’s coming in, **analyze** using expert knowledge, and **respond** by deciding on and applying good practices.
- **Scrum** can certainly work in a complicated domain (e.g., software maintenance), but it might not be the best solution.

  - Analysis and design processes based on expert knowledge and good practices may be better options.
Cynefin: Simple (Obvious) Domain

- In simple (obvious) problems, the domain is stable and everyone can see cause and effect.
- Often the right answer is obvious and undisputed.
- This is the domain of legitimate best practices; there are known solutions.
- Once we **assess** the facts of our situation and **categorize** them, we can **respond** by determining and applying the proper predefined solution.
- **Scrum** can be used in a simple (obvious) domain, but it may not be the most efficient tool.
  - Using a process with a well-defined, repeatable set of steps that are known to solve the problem would be a better fit.
  - For example, if we want to reproduce the same product over and over again, a well-defined assembly-line process would be a better fit than Scrum.
Cynefin: Chaotic Domain

- In chaotic problems, there is a crisis and we need to act immediately to prevent further harm and reestablish at least some order.
- There is no clear cause and effect.
- There are many decisions to make, and no time to think.
- No one knows the answers.
- We have to look for what works instead of the right answers.
- We need to act immediately, then inspect to see if the situation has stabilized, and then adapt to migrate the context to the complex domain.
- **Scrum** is not the best solution in a chaotic domain.
  - In such domains, we are not interested in prioritizing a backlog of work and determining what work to perform in the next iteration. We need to act fast.
Cynefin: Disorder

- You are in the disorder domain when you don’t know which of the other domains you are in.
  - This is a dangerous place to be because you don’t know how to make sense of your situation.
  - In such cases, people tend to interpret and act according to their personal preferences for action; this is often a recipe for disaster.

- When in the disorder domain, the way out is to break down the situation into constituent parts and assign each to one of the other four domains.

- You are not trying to apply **Scrum** in the disorder domain; you are trying to get out of this domain.
Interrupt-Driven Contexts

- **Scrum** is not well suited to **highly interrupt-driven** (or request-driven) work.
  - Interrupts may disrupt your plans and constantly change your priorities, prohibiting reliable planning of iterations of a week or more.
  - For example, during product support, the product backlog is populated continuously as you receive support requests.

- In interrupt-driven environments you would be better off considering an alternative agile approach called **Kanban**.
  - Kanban is not a stand-alone process solution, but instead an approach that is overlaid on an existing process.
Cynefin and Software Development

- The many facets of software development and support will not fit nicely into just one Cynefin domain.

- Most software development work falls in the domains of complicated or complex.

- However, software development is a rich endeavor, with aspects that overlap and activities that fall into all of the different domains.

  - It should be noted that the spectrum of work can range from innovative new-product development to maintenance/support.
References
