HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

h_da

Agile Software Development

Part 3: Adaptive Software Development

Prof. Dr. A. del Pino

Adaptive Software Development

Planning

Estimating and planning are not just about determining an appropriate deadline or schedule. Planning – especially an ongoing iterative approach to planning – is a quest for value. Planning is an attempt to find an optimal solution to the overall product development question: What should we build ?

Source: M. Cohn. *Agile Estimating and Planning*. Prentice-Hall, 2006

Adaptive Software Development

Iterative Life Cycle vs. Adaptive Development Life Cycle

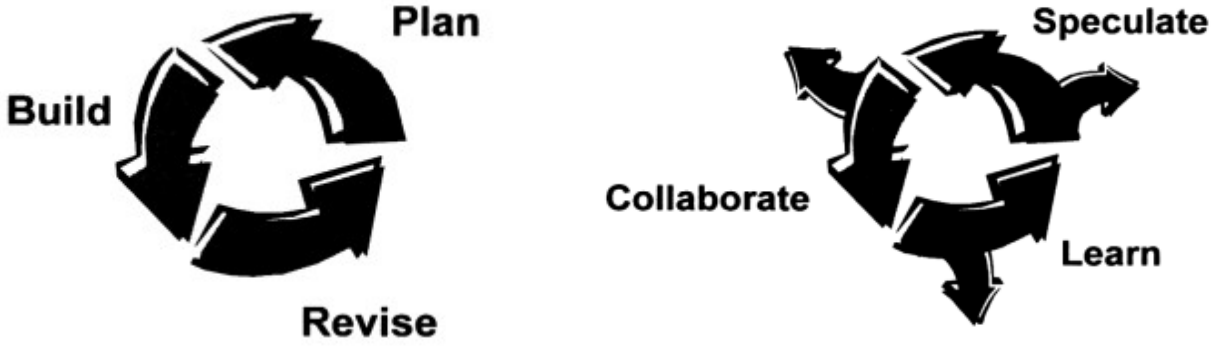

Image source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 39,41

> The secondary arrows [...] leading away from the iterative circle, represent breakout ideas that identify results far afield from the original project mission profile. Adaptive projects, like adaptive organizations, are open to possibility – one never knows where the next innovative product will take seed.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 40

Adaptive Software Development

Development tasks span over several adaptive cycles

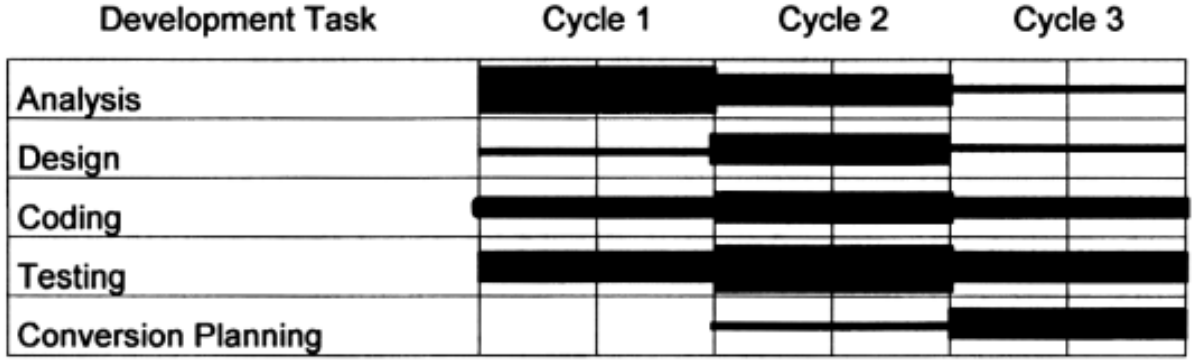

Image source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 100

One reason the adaptive approach seems so different is that in traditional sequential life cycles, analysis or data design activities are performed during the corresponding life-cycle phases. In an adaptive approach, these activities will be performed during several cycles. There may for example, be more analysis effort in early cycles, but analysis will occur in all cycles.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 99

Adaptive Software Development

Deviations

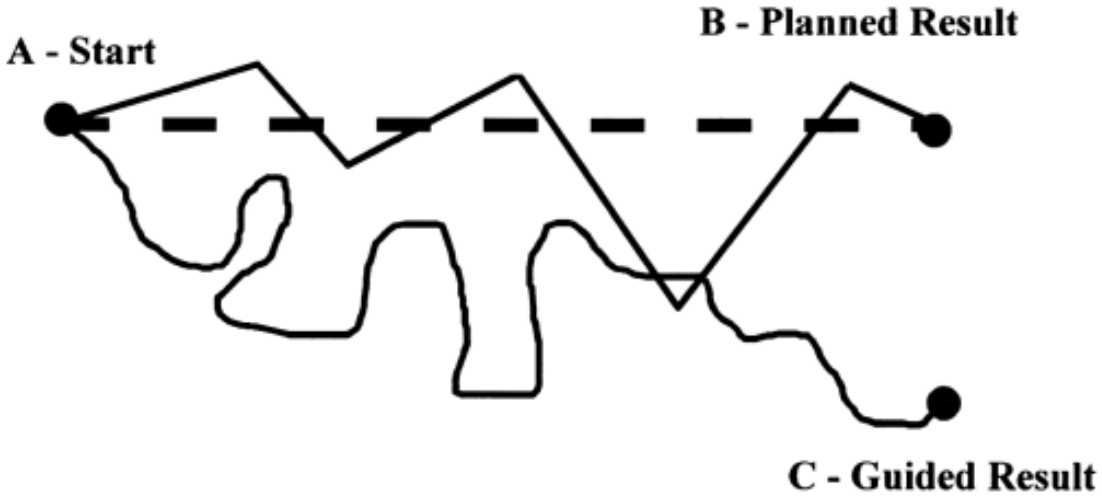

Image source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 43

Traditional approaches:

> Deviations from plan are mistakes that must be corrected.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 42

Adaptive approach:

> Deviations guide us toward the correct solution.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 43

Adaptive Software Development

The Adaptive Life Cycle

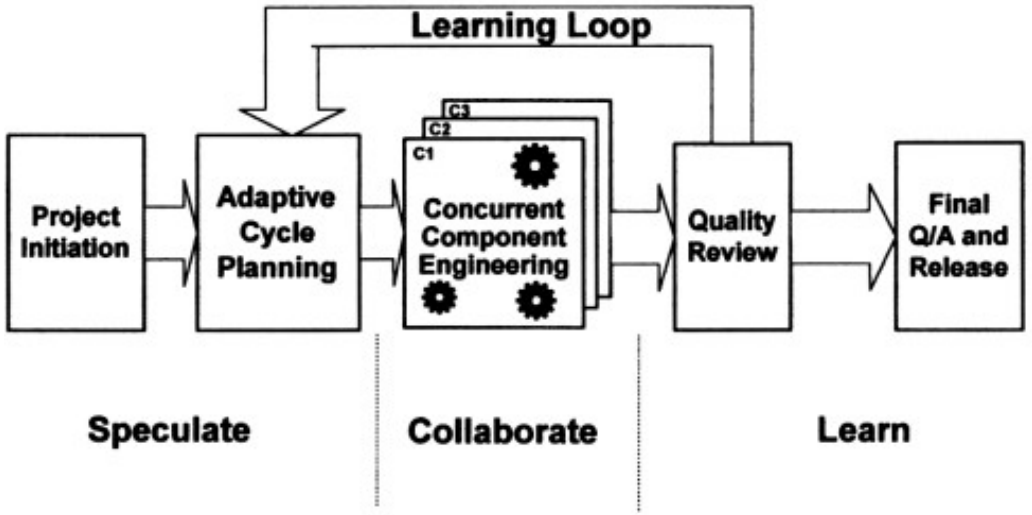

Image source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 84

Five characteristics:

- Mission-Driven
- Component-Based
- Iterative, converging
- Time-Boxed
- Risk-Driven, change-tolerant

Adaptive Software Development

Adaptive Planning – Versions, Cycles, Builds

A version is a product that is ready to be installed or, as in the case of shrink-wrapped software, to be manufactured.

[...]

A cycle delivers a demonstrable portion, or component, of the product to a review process.

[...]

A build constructs an interim portion of the product and makes the product visible to the development team.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 42

Adaptive Software Development

Step 1 – Conduct a Project Initiation Phase

> If the project does not start off well, it may flounder, sometimes for months. The objective of the project initiation phase is to clearly establish project expectations among all the project's stakeholders. The project initiation phase is the stage at which mission statements [...] would be produced.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 93

What are the results of this phase from the perspective of (a) the project sponsor; (b) the customer; and (c) the developers ?

Adaptive Software Development

Step 2 – Determine the Project Time-Box

> The project time-box specifies the estimated duration of the project. [...] given the uncertainty of complex projects, estimates may not prove to be particularly accurate. The project time-box should be viewed as a boundary, not as a goal.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 94

What do we know about software estimates ? ...

Adaptive Software Development

A few facts about software estimates

- Most software estimates are performed at the beginning of the life cycle. This makes sense until we realize that estimates are obtained before requirements are defined and thus before the problem is understood. Estimation, therefore, usually occurs at the wrong time.
- Most software estimates are made either by upper management or by marketing, not by the people who will build the software or their managers. Estimation is, therefore, done by the wrong people.

Source: Robert L. Glass. *Facts and Fallacies of Software Engineering*. Addison-Wesley, 2003, pg. 31 ff.

Adaptive Software Development

Step 3 – Determine the Cycles

> The time allotted a cycle should be on the order of four-to-eight weeks for projects of less than nine months and six-to-ten weeks for projects longer than nine months. [...] Early cycles should usually be shorter than later cycles to encourage customer involvement.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 94

Adaptive Software Development

Cycle duration vs. Uncertainty

> [Cycle duration] should reflect the development team's perception of the uncertainty about the product's requirements. Shorter cycles should be used for areas of high uncertainty; longer cycles should be used for areas of greater certainty. A single cycle, essentially one using a waterfall approach, is appropriate where the environment and the project's requirements are relatively stable.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 95

Adaptive Software Development

Step 4 – Create an objective statement for each cycle

> A detailed task list is no substitute for a clear, focusing statement – a cycle objective statement [...]. The detailed task list just insures that everyone will be busy not necessarily productive. The cycle objective statement provides the next level of definition of the project's mission.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 96

Adaptive Software Development

Example Situation

Assume the following situation:

- Customer involvement is hard to establish, and is often perfunctory when achieved.
- Scope issues are difficult to understand and resolve.
- Technical issues are less important to the customer than business issues.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 96

Adaptive Software Development

...And its Resulting Cycle 1 Objective Statement

The corresponding objective statement for the first cycle might look like this:

> Ensure overall project viability by actively engaging the customer in the project while confirming the feature scope, schedule, and resource estimates.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 96

> Note, until cycle 1 has been completed, nothing concrete will have been produced. [...] The first cycle can be thought of as a *proof of concept*.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 96

Adaptive Software Development

Step 5 – Assign components to cycles

> Once the overall objective of each of the cycles has been determined, core team members will assign components to the most appropriate cycle. A component should be assigned to the cycle in which the bulk of its development is to be accomplished [...].

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 97

Adaptive Software Development

Step 6 – Assign technology and support components

A software development project entails much more than executable modules. Besides the code-related components of the product, a project also must produce documentation and install required technology components. Both are costly and time-consuming.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 98

Adaptive Software Development

Step 7 – Create a project task list (1)

This step involves two options. The first alternative is to eliminate the step completely. One of the tenets of adaptive management is to define the desired results, the "what," and let the development team figure out the details of delivering the results, the "how." [...]

Managers experienced in more traditional project management techniques will feel uncomfortable with this "loose" approach.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 99

Sounds good, but how to monitor progress with this appoach ?

Adaptive Software Development

Step 7 – Create a project task list (2)

The second alternative for this step is to begin with the task list created from the components and add additional detail-level tasks. [...]

Developing a project task list is often necessary in the transition to an adaptive approach. It provides some comfort to project managers and team members who have concerns about a less structured approach.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 99

Adaptive Software Development

Cycle Planning

> The best strategy is to loop through the steps several times in the planning process. In numerous planning sessions with clients, my experience has been that, in the process of assigning components to cycles, the team gains a much better understanding of both the components and the overall project mission. The assignment process raises issues of project scope, component definition, component priorities and dependencies, size estimates, resource needs and availability and more.
>
> Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 99

Which general principle stands behind this approach ?

Adaptive Software Development

Cycle Reviews: Feedback Loops vs. Learning Loops

Reviews at the end of each cycle should answer the following questions:

- Is the project on track if viewed from a broad scope, schedule, defect-level, and resource perspective? [...]
- Are the project mission artifacts still valid?
- Does the quality of each delivered component meet customer and technical specifications and expectations?
- Is the project team working efficiently and effectively?

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 101

Adaptive Software Development

Cycle Reviews

In traditional projects, tracking schedules, tasks, and resources usually takes precedence over tracking components and evaluating their quality characteristics. In adaptive projects, tracking of components and their quality takes precedence. Built on the principle that the application is the only acceptable model to the customer, adaptive development projects track the delivery of actual software features rather than documents.

Source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 101

Deja vu ?