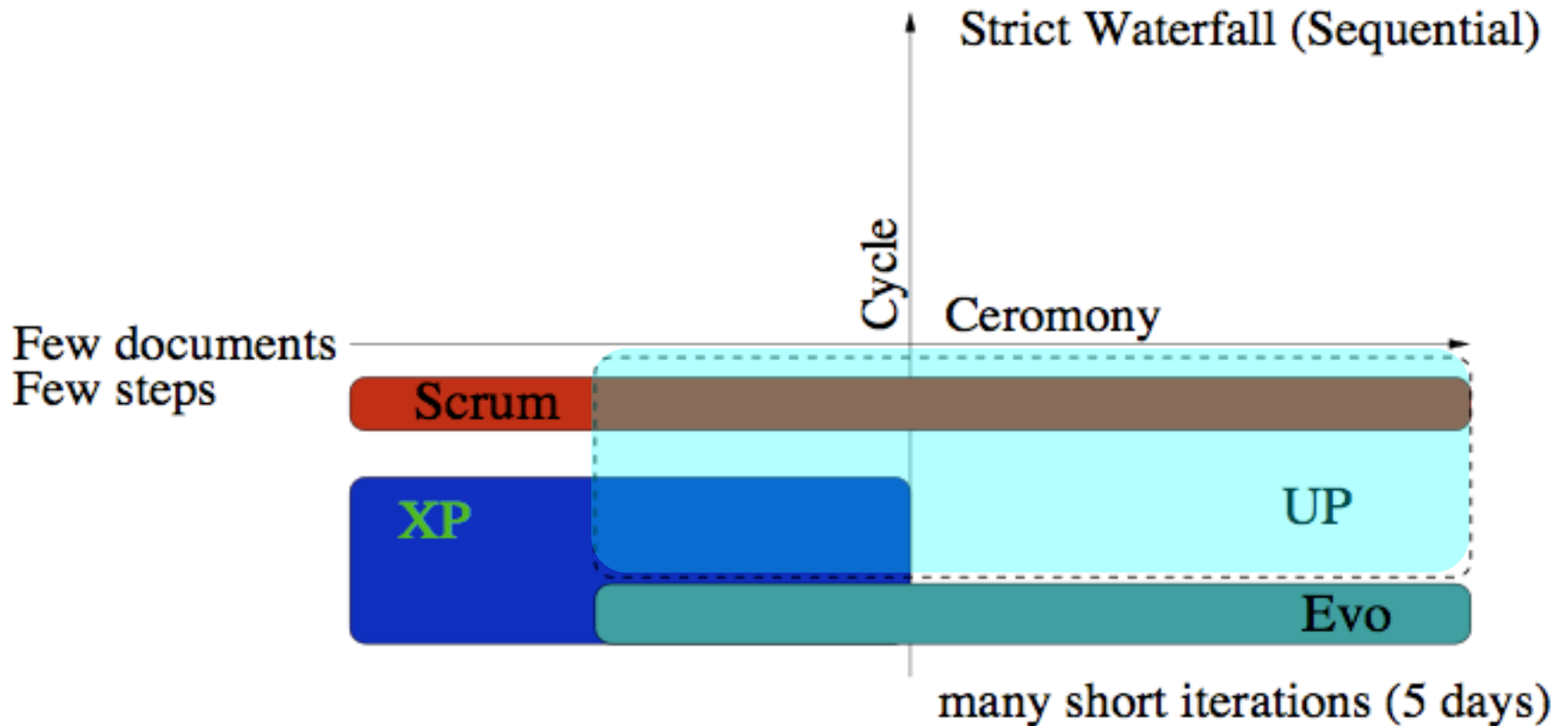


Evidence and Experience

Peter Dolog
dolog [at] cs [dot] aau [dot] dk
E2-201
Information Systems
February 27, 2007





source: Agile & Iterative Development, C. Larman

Short Comparison

SCRUM:

Simple practices, self management, daily scrum, , product owner, product backlog, demo at the end of each iteration, 30days l., no documents defined

UP:

2-6 weeks iterations, guidelines for many artifacts, risk centered, value oriented and with vision, architecture, use cases, sometimes incorrectly applied as waterfall, minimal attention to social dynamics, impression of predictable development

XP:

Pair programming, testing, team code ownership, sustainable pace, coding standards, small frequent releases, 1-3 weeks iterations, requires presence of customer, relying on history, no standards for specifications

EVO:

5 days iterations with delivery at the end, client and value driven, quality requirements with numeric measures, own language for specification

Evidence

Larman's Chapter 6

Project Duration and Team Size

... Directly Affect Project Success

Project Size	People	Time (months)	Success Rat
Less than \$750K	6	6	55%
\$750K to \$1.5M	12	9	33%
\$1.5M to \$3M	25	12	25%
\$3M to \$6M	40	18	15%
\$6M to \$10M	+250	+24	8%
Over \$10M	+500	+36	0%

Jim Johnson: **Turning Chaos into Success**, SoftwareMag (December 1999)

Growing vs. Developing Software

We have long been convinced that shorter time frames, with delivery of software components early and often, increase the success rate. Shorter time frames foster an iterative process of design, prototype, develop, test, and deploy small elements. "Growing" (instead of "developing") software engages the user earlier and confers ownership. And because each software component has a clear and precise statement and set of objectives, realistic user expectations are set.

Jim Johnson: **Turning Chaos into Success**, SoftwareMag (December 1999)

Size research

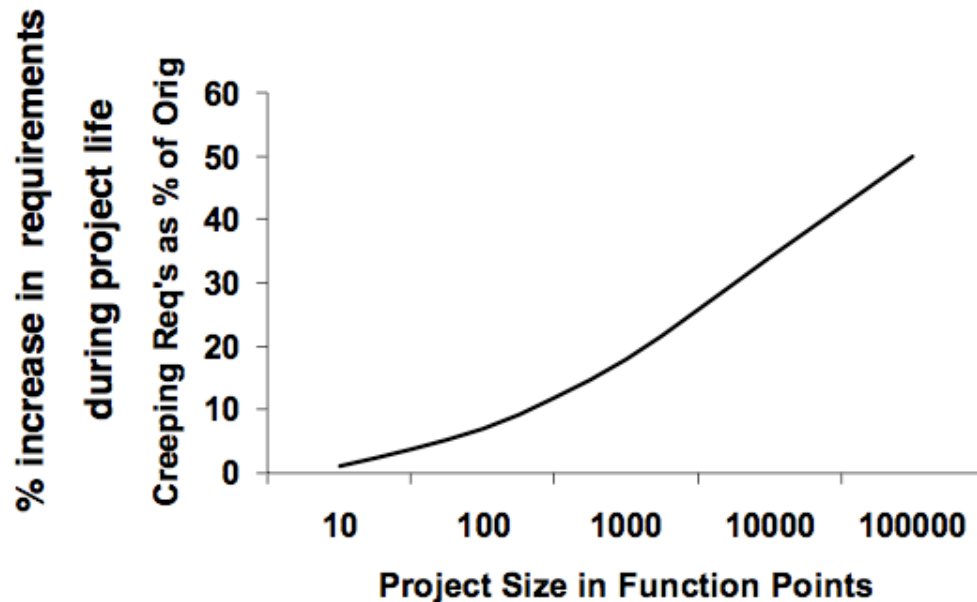
It is clear from the research that the smaller a project, the less likely it is to fail: 47.4% of successful projects were completed in six months of elapsed time and another 42.1% were finished in 12 months. This means 89.5% of successful projects were completed in 12 months.

Among the successful projects, 35% required less than 24 person months. Most (65%) were completed within 48 person months. Among the unsuccessful projects, by contrast, 56.8% required 72 person months or more.

This is not to say that projects over 12 months should not be started but that they should be broken into smaller projects within a programme of change whenever possible.

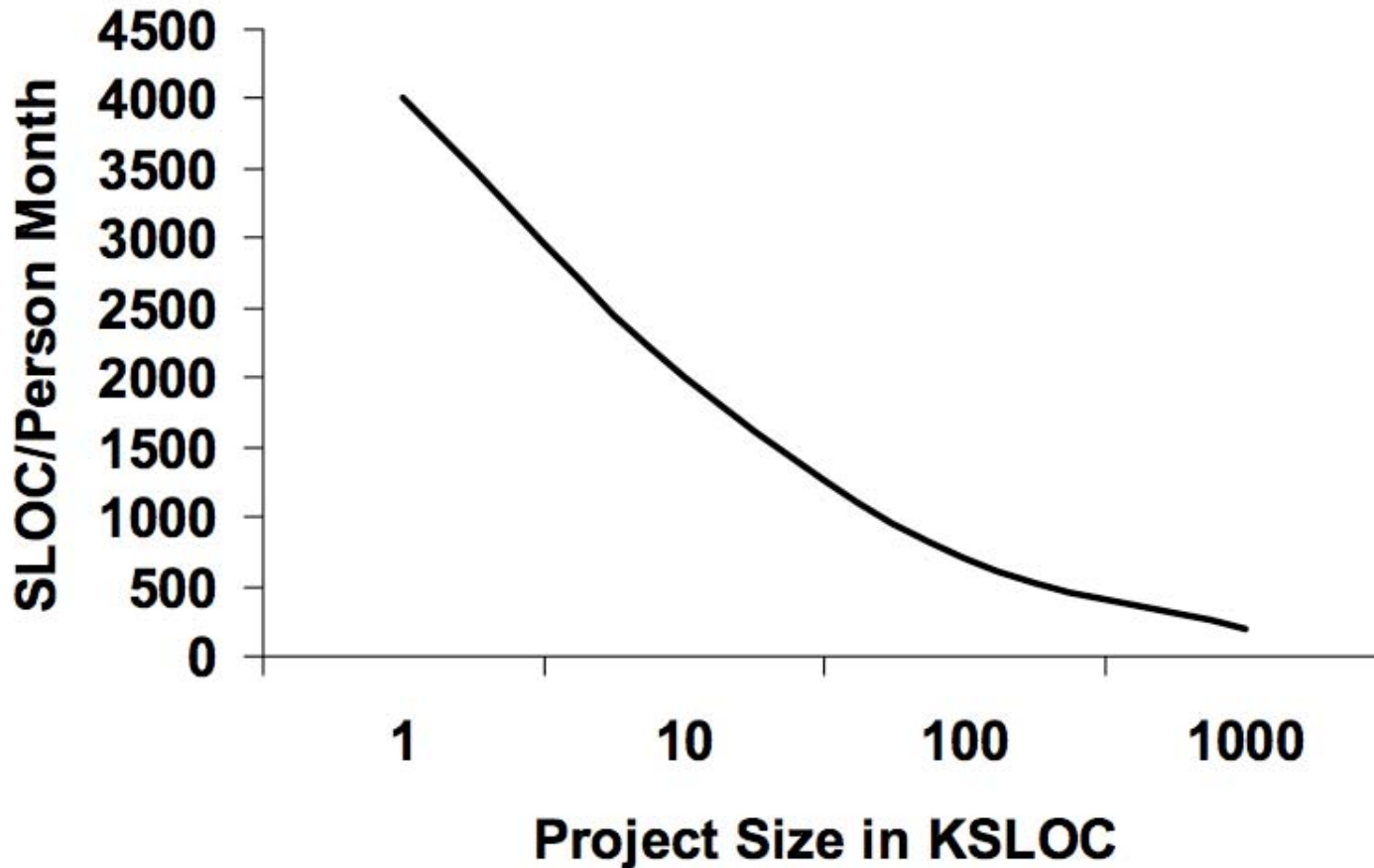
Rates of Change on SW Projects

Growth in requirements



Source: Applied Software Measurement, Capers Jones, 1997. Based on 6,700 systems.

Complexity vs. Productivity



Source: Measures For Excellence, Putnam, 1992.
Based on 1,600 systems.

From a two-year study

"Now there is proof that the evolutionary approach to software development results in a speedier process and higher-quality products."

"The most striking result to emerge from the research concerned the importance of getting a low-functionality version of the product into customer's hands at the earliest opportunity. The differences in performance are dramatic. That one parameter explains more than one-third of the variation in product quality across the sample—a remarkable result."

A. MacCormack, "Product-Development Practices That Work,"
MIT Sloan Management Rev., vol. 42, no. 2, 2001, pp. 75-84

Practices for Success

MacCormack points to four development practices that spell success:

An early release of the evolving product design to customers

Daily incorporation of new software code and rapid feedback on design changes

A team with broad-based experience of shipping multiple projects

Major investments in the design of the product architecture

A. MacCormack, “Product-Development Practices That Work,”
MIT Sloan Management Rev., vol. 42, no. 2, 2001, pp. 75-84

Productivity: Patterns of success

Iterative development

Simple organizational structure; fewer roles than average.

Architect worked as programmer

High verbal communication among the team; a technical synch-and-discuss meeting each day

Small tiger-team built the core architecture first

Harrison, N, and Coplien, J. 1996. "Patterns of Productive Software Organizations." *Bell Labs Technical Journal*, Summer 1996

Shine Survey 2003

In these cost conscious times, it was amazing to find that 95% of respondents believed that costs were the same or less. Many companies may find this reason enough to trial the use of Agile processes in their business. Once they do, they will find that the benefits go way beyond cost to deliver stunning improvements in productivity, quality and business satisfaction.

<http://www.shinotech.com/display/www/Extreme+success+with+Agile>

Shine Survey 2003

93% said team productivity improved

88% found the quality of applications was better

83% experienced better business satisfaction with the software

Shine Survey 2003

Question 1: How would you rate your understanding of Agile Methodologies?	Very Limited, Limited, Average, Extensive, Very Extensive
Question 2: What form of Agile processes are you most using at the moment?	Xtreme, Scrum, Crystal, Feature Driven, Other
Question 3: Has adoption of Agile processes altered your team productivity?	Much worse, Somewhat worse, Unchanged, Better, Significantly better
Question 4: Has adoption of Agile processes altered the quality of your applications?	Much worse, Somewhat worse, Unchanged, Better, Significantly better
Question 5: Has adoption of Agile processes altered the cost of development?	Much more expensive, More expensive, Unchanged, Less expensive, Much less expensive
Question 6: Has adoption of Agile processes altered the level of business satisfaction with the software?	Much worse, Somewhat worse, Unchanged, Better, Significantly better
Question 7: What feature of your Agile processes do you like the most?	People over processes, Code over documentation, Relationships over contracts, Respond to change over plan, Other
Question 8: What feature of your Agile processes makes you most uncomfortable?	Low documentation, Lack of planning, Lack of project structure, Lack of authority, Other
Question 9: Do you intend to use or adopt Agile processes in the next year?	Yes, No
Question 10: What proportion of projects do you believe are appropriate for Agile processes?	0% (All), 25% (Some), 50% (Half), 75% (Most), 100% (All)

Radical Collocation

Companies are experimenting with putting teams into warrooms, hoping for some productivity enhancement.

We conducted a field study of six such teams, tracking their activity, attitudes, use of technology and productivity.

Teams in these warrooms showed a doubling of productivity.

Table 1. Comparative statistics on productivity measures.

	Pilot Teams	Company Baseline	Industry Standard
Function points per staff month (higher is better)	29.49	14.35	20.00
Cycle Time (lower is better)	7.64	19.47	24.00

Table 2. Satisfaction measures for the pilot teams.

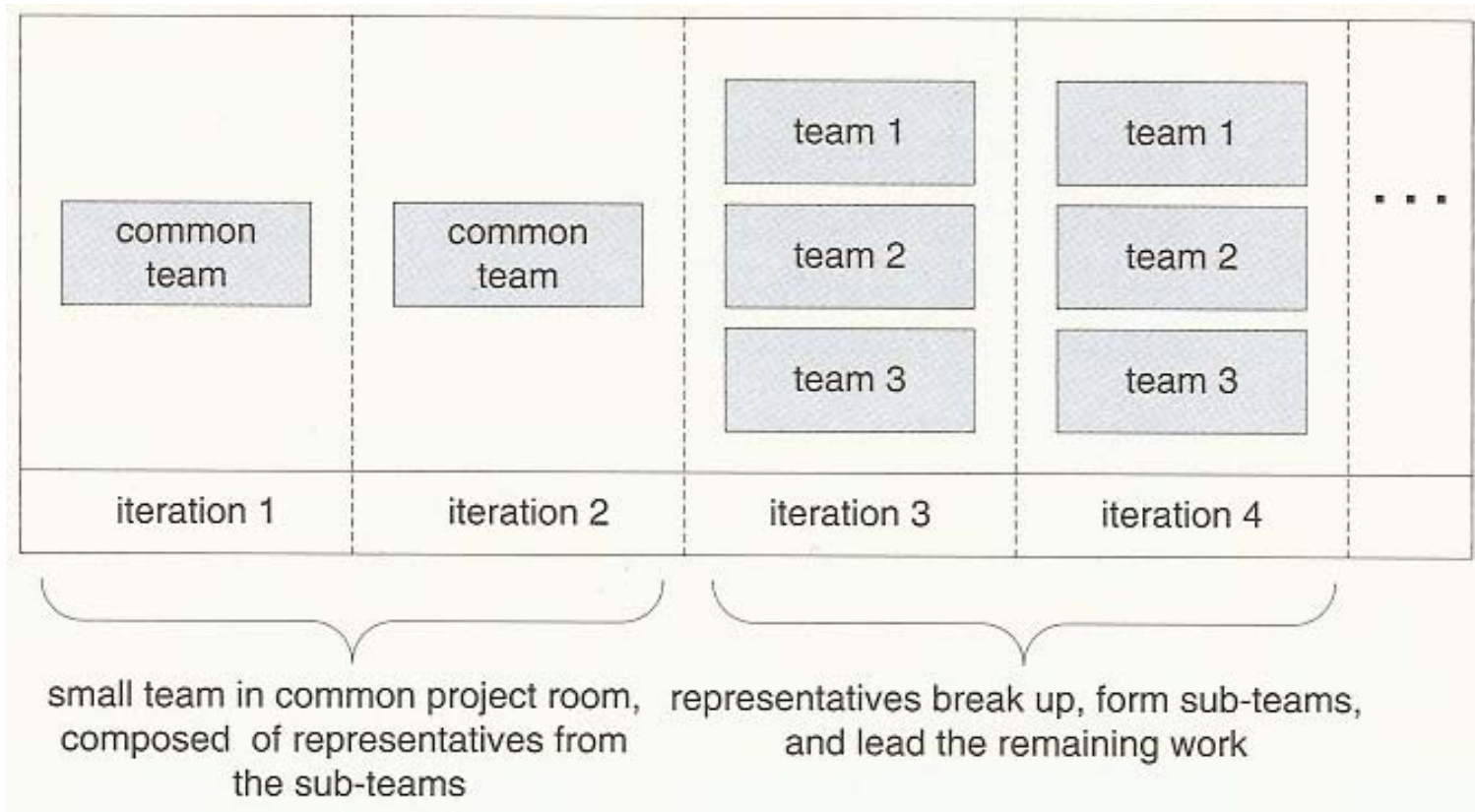
Team Satisfaction	4.15
Sponsor Satisfaction	4.56
End User Satisfaction	3.68

Teasley, S., Covi, L., Krishnan, M. S., and Olson, J. S. 2000. How does radical collocation help a team succeed?. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, Pennsylvania, United States). CSCW '00. ACM Press, New York, NY, 339-346

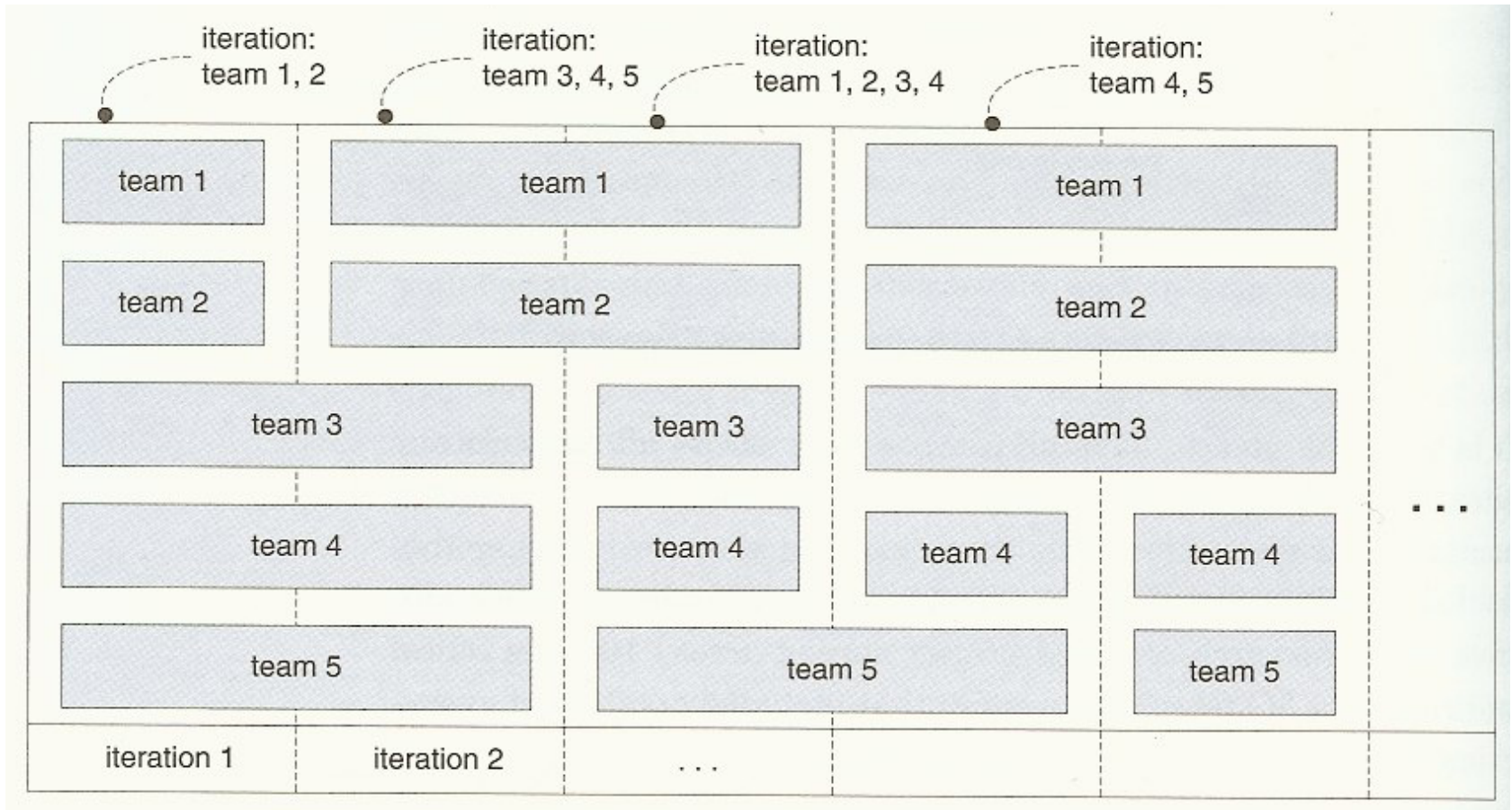
Practice

Larman Ch. 11

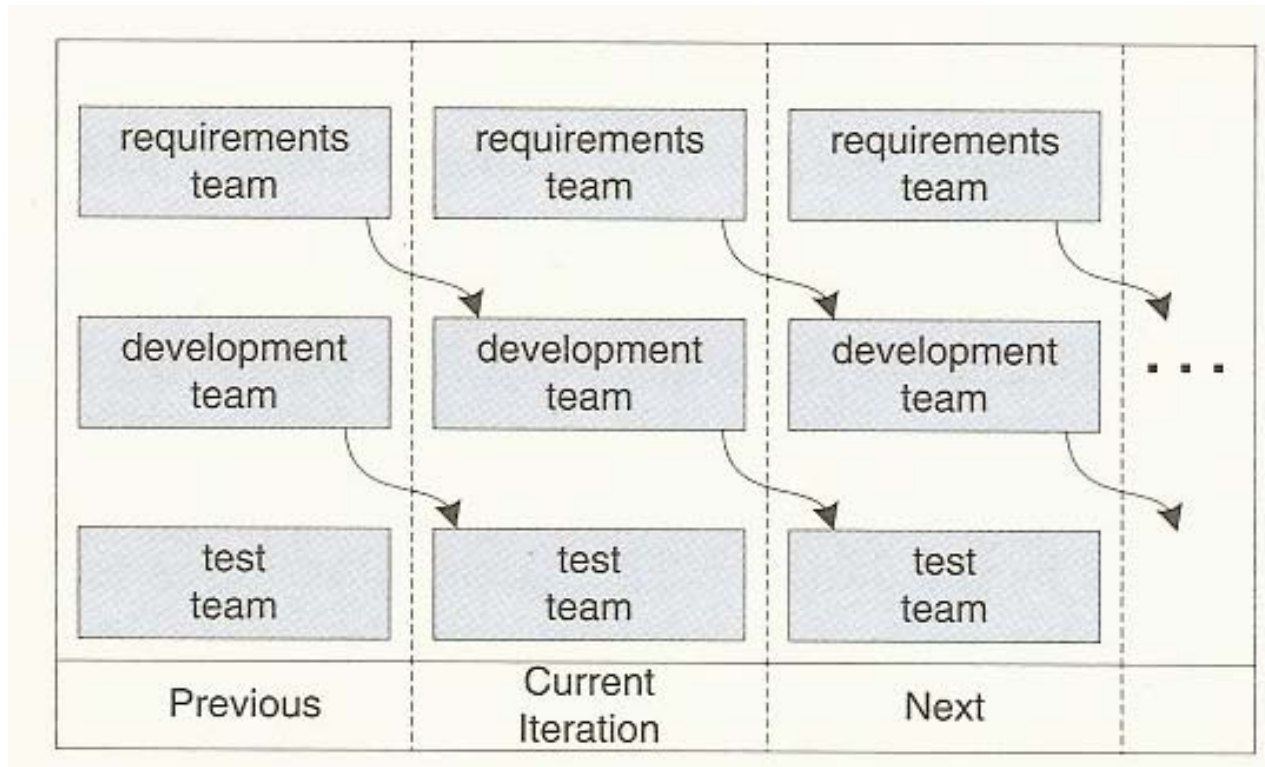
11.1 Multiteam Development



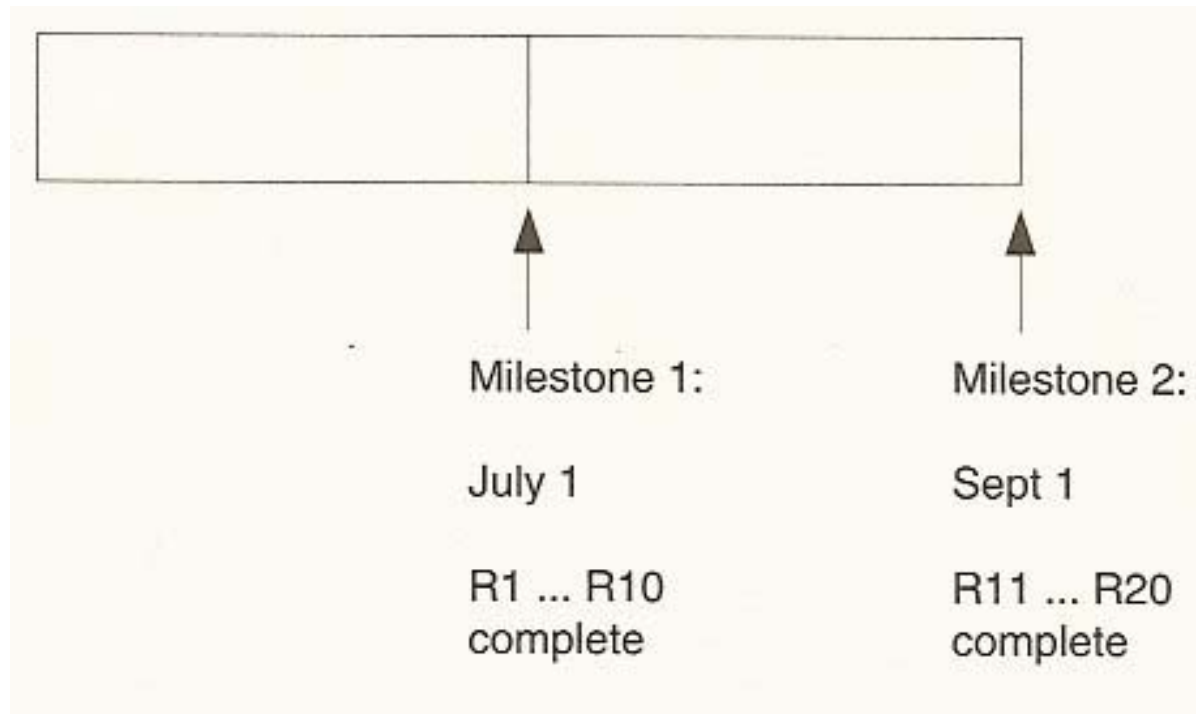
11.2 Sub-team Iteration



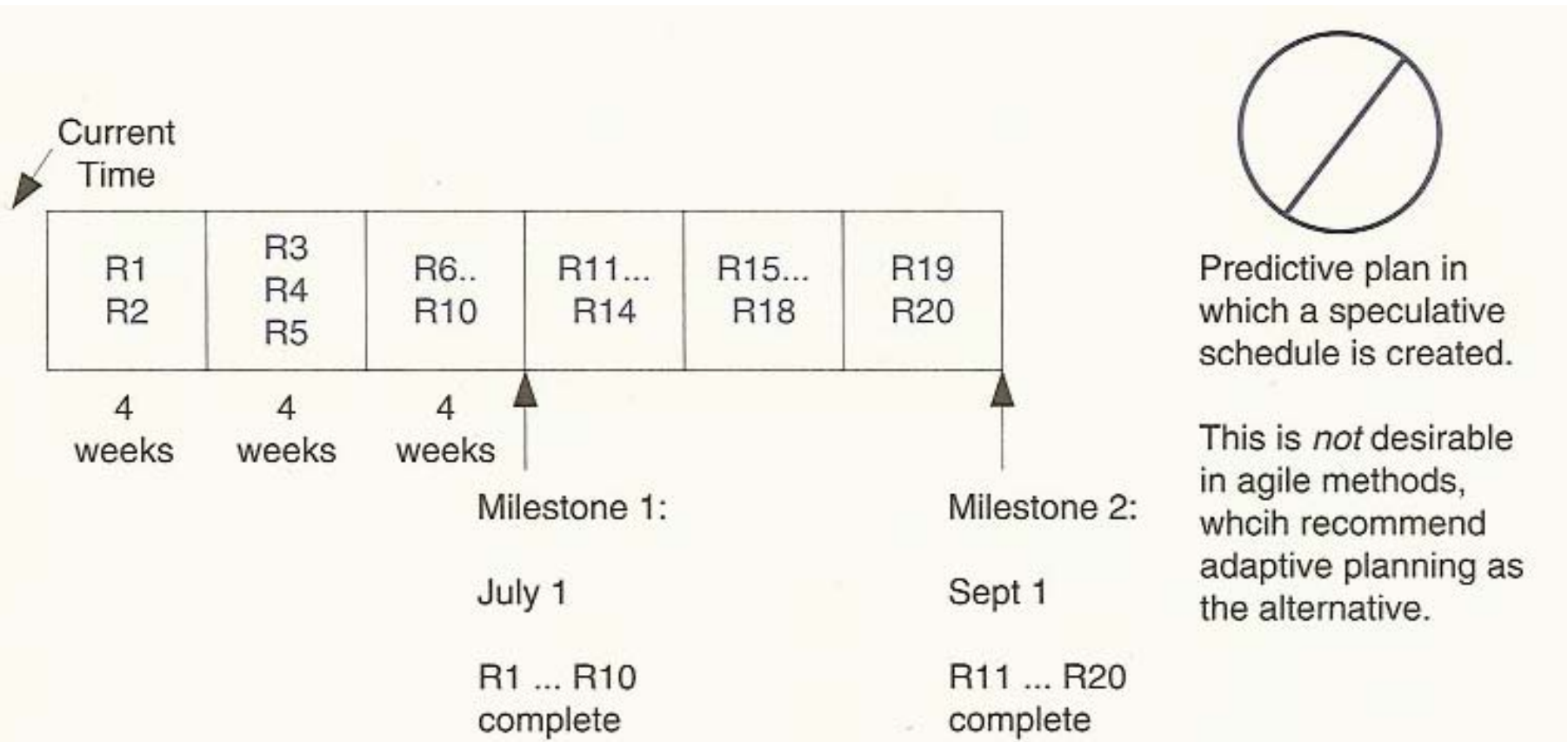
11.3 Pipelining Iteration Work



11.4 Milestones estimated



11.5 Predictive Plan



11.6 Adaptive Plan

