Software Process

Dr Halil Ali

Lecturer in Software Engineering

Week 2: Lectorial

Lecture 2: Software Process

Key Points

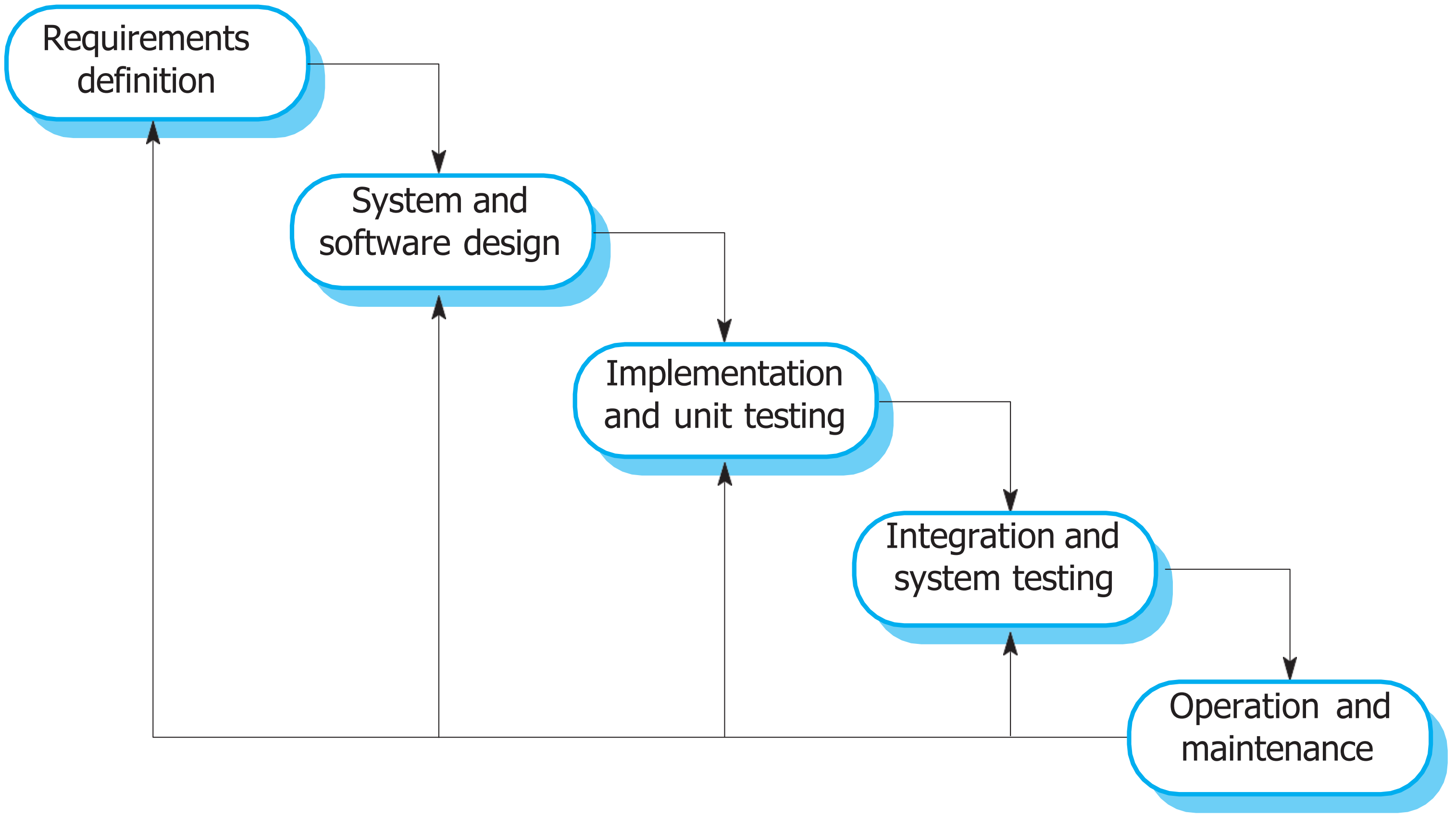
Software process

- A structured set of activities is required to develop a software system.
- There are many different software processes
  - but all involve:
  - **Specification** – defining what the system should do;
  - **Design and implementation** – defining the organization of the system and implementing the system;
  - **Validation** – checking that it does what the customer wants;
  - **Evolution** – changing the system in response to changing customer needs.

Software processes define the ***who***, ***what***, ***when***, and ***how*** of developing software.

For example: A software process should specify the activities (***how***) a tester (***who***) should do, the order of these activities (***when***), and the artifacts (***what***) that should be produced by a tester.

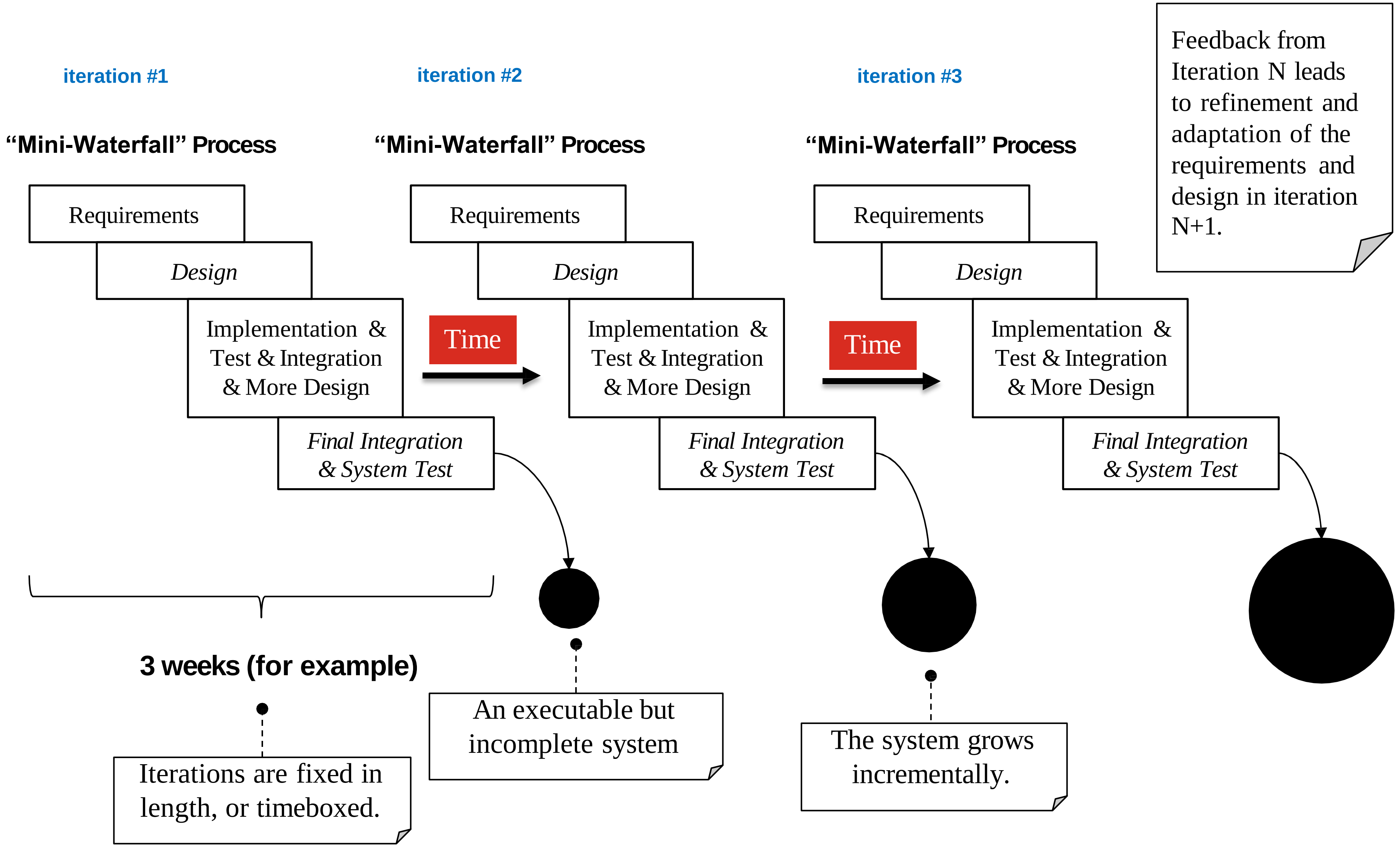
The waterfall model


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Question

Name some of the limitations of the Waterfall Model

Iterative and Incremental Development (IID)



Feedback from Iteration N leads to refinement and adaptation of the requirements and design in iteration N+1.

Iterations are fixed in length, or timeboxed.

An executable but incomplete system

The system grows incrementally.

Image Source: Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition

Requirements changes and the project size

- Larger software projects experience more changes in requirements

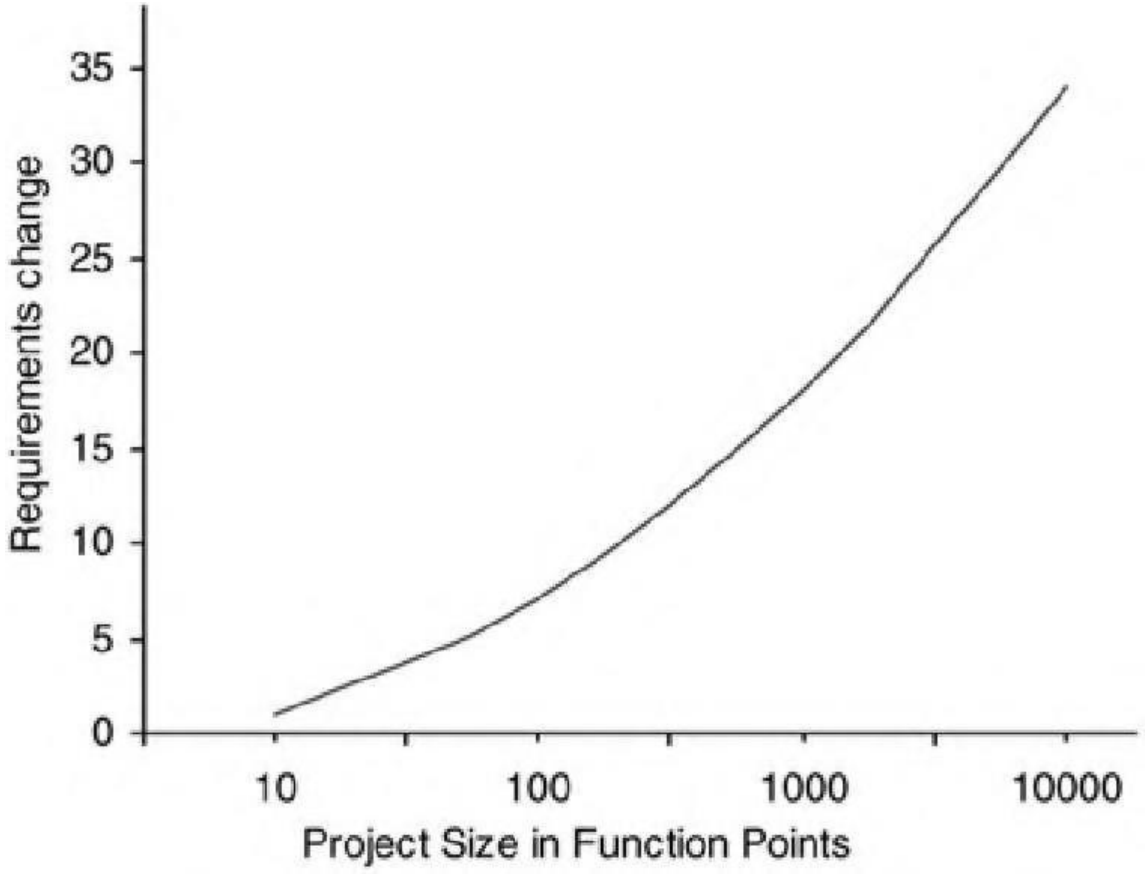


Image Source: Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition

Unified Process (UP) Workflows and Phases

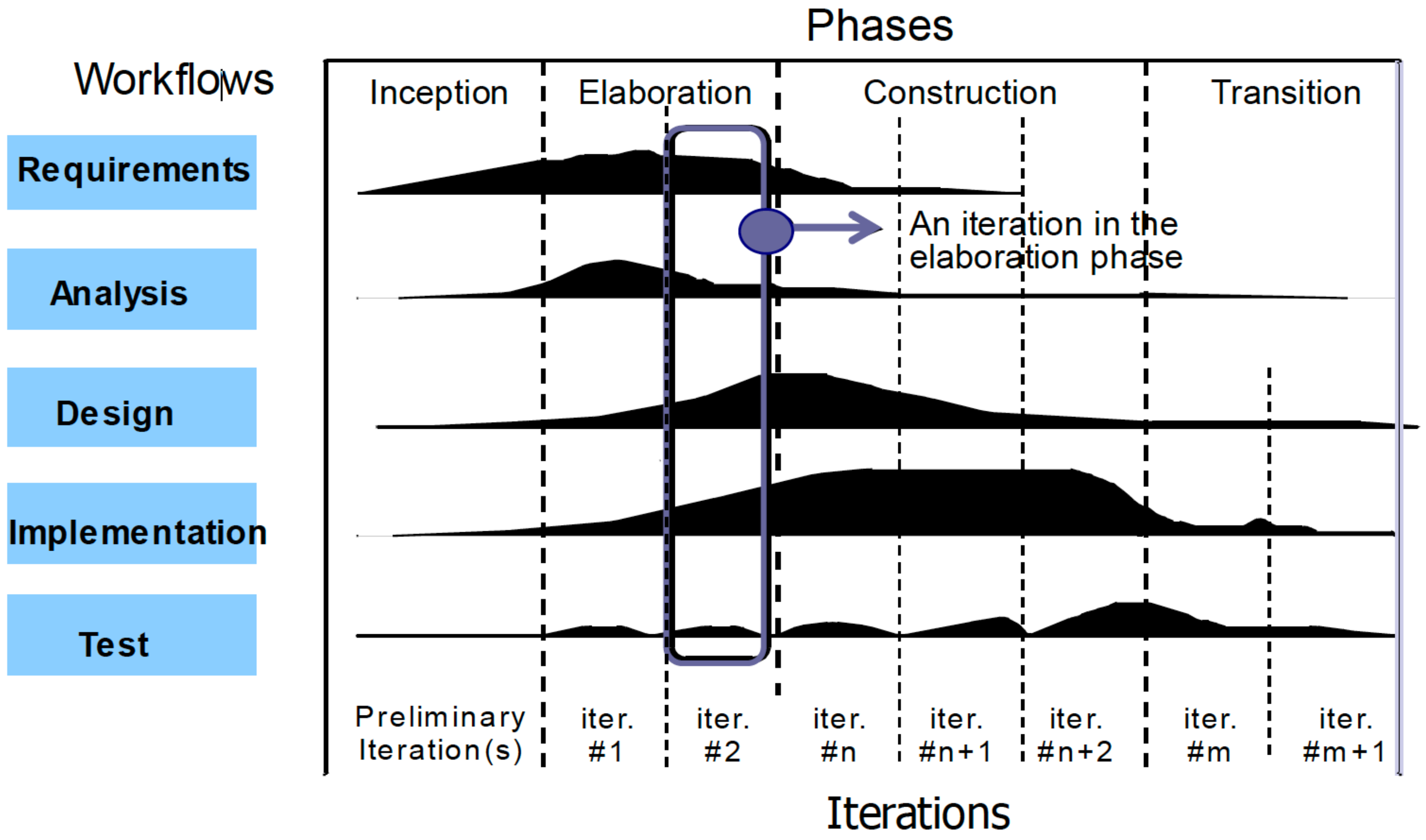


Image source: http://sharif.edu/~ramsin/index_files/undergradcourse_OOD.htm

Efforts across the disciplines

- During one iteration work goes on in most or all workflows
- The relative effort across these workflows changes over time.
- Early iterations naturally tend to apply greater relative emphasis to **requirements** and **design**, and later ones less so, as the requirements and core design stabilize through a process of feedback and adaptation.

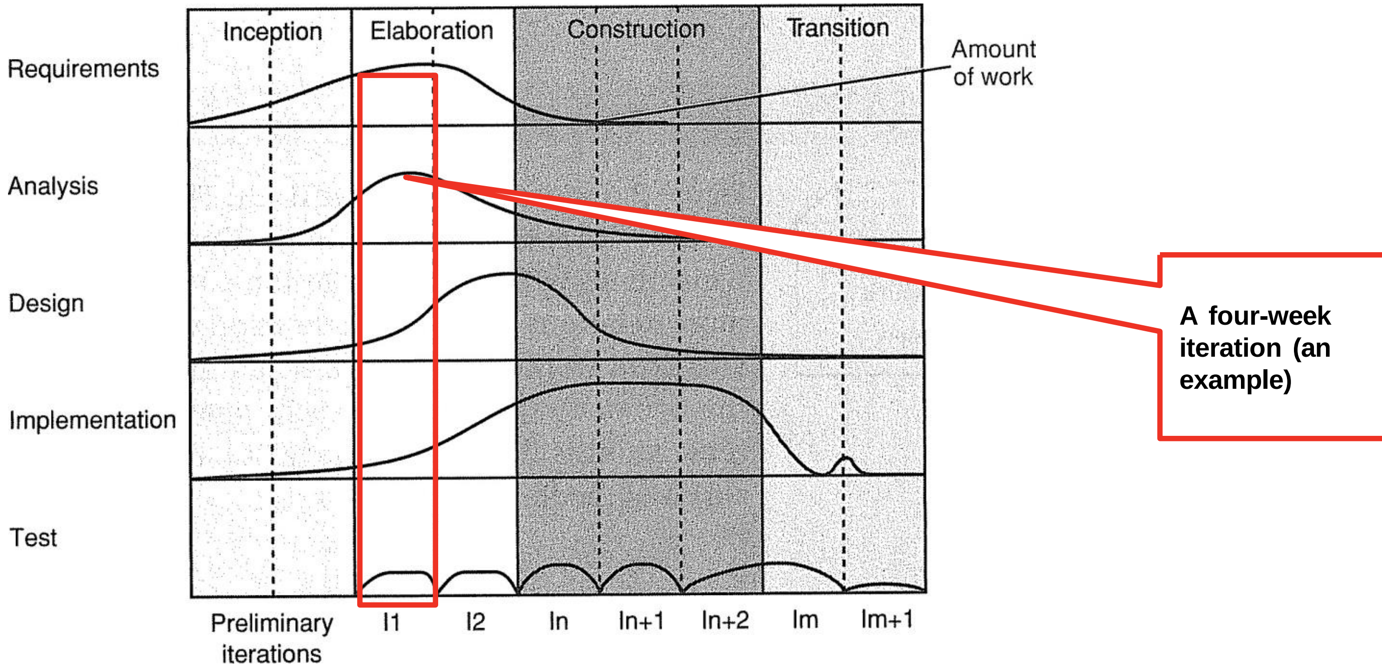
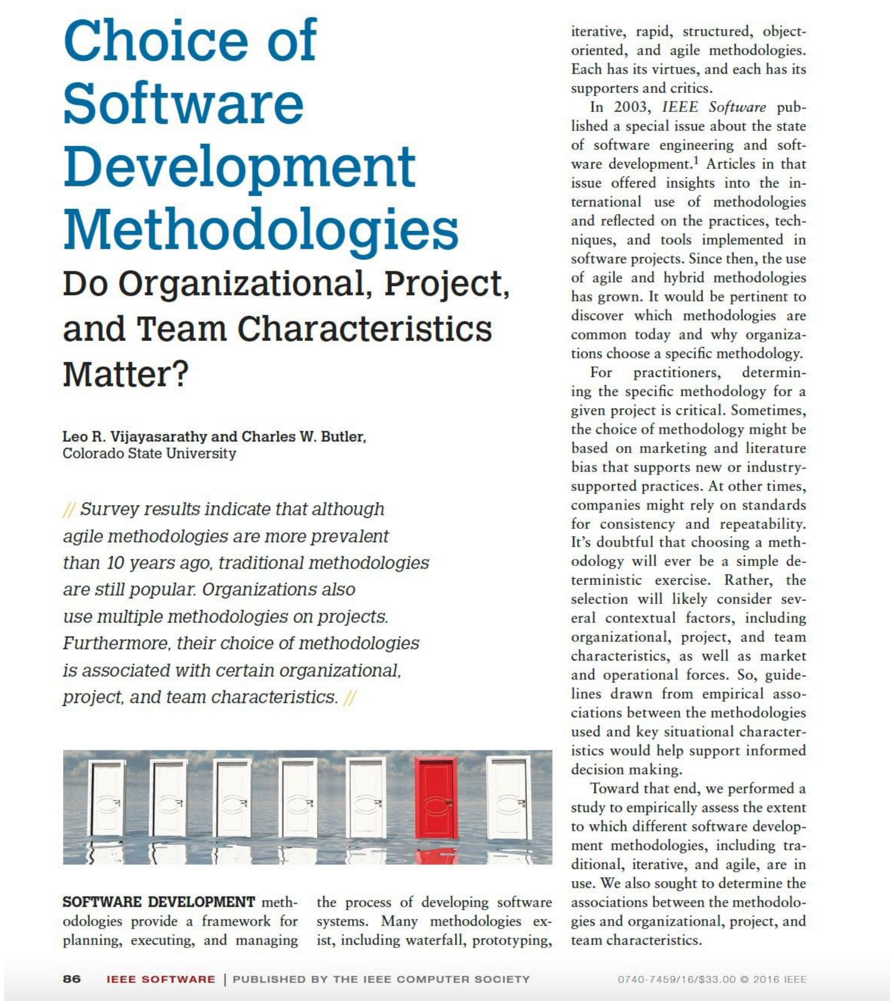


Image source: Arlow, J., Neustadt, I., UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design, 2nd Ed. Addison-Wesley, 2005.

Software Processes in the Software Industry



Choice of Software Development Methodologies

Do Organizational, Project, and Team Characteristics Matter?

Leo R. Vijayasarathy and Charles W. Butler,
Colorado State University

// Survey results indicate that although agile methodologies are more prevalent than 10 years ago, traditional methodologies are still popular. Organizations also use multiple methodologies on projects. Furthermore, their choice of methodologies is associated with certain organizational, project, and team characteristics. //

SOFTWARE DEVELOPMENT methodologies provide a framework for planning, executing, and managing the process of developing software systems. Many methodologies exist, including waterfall, prototyping, iterative, rapid, structured, object-oriented, and agile methodologies. Each has its virtues, and each has its supporters and critics.

In 2003, *IEEE Software* published a special issue about the state of software engineering and software development.[1] Articles in that issue offered insights into the international use of methodologies and reflected on the practices, techniques, and tools implemented in software projects. Since then, the use of agile and hybrid methodologies has grown. It would be pertinent to discover which methodologies are common today and why organizations choose a specific methodology.

For practitioners, determining the specific methodology for a given project is critical. Sometimes, the choice of methodology might be based on marketing and literature bias that supports new or industry-supported practices. At other times, companies might rely on standards for consistency and repeatability. It's doubtful that choosing a methodology will ever be a simple deterministic exercise. Rather, the selection will likely consider several contextual factors, including organizational, project, and team characteristics, as well as market and operational forces. So, guidelines drawn from empirical associations between the methodologies used and key situational characteristics would help support informed decision making.

Toward that end, we performed a study to empirically assess the extent to which different software development methodologies, including traditional, iterative, and agile, are in use. We also sought to determine the associations between the methodologies and organizational, project, and team characteristics.

86 IEEE SOFTWARE | PUBLISHED BY THE IEEE COMPUTER SOCIETY 0740-7459/16/$33.00 © 2016 IEEE

A Research Study in 2016*

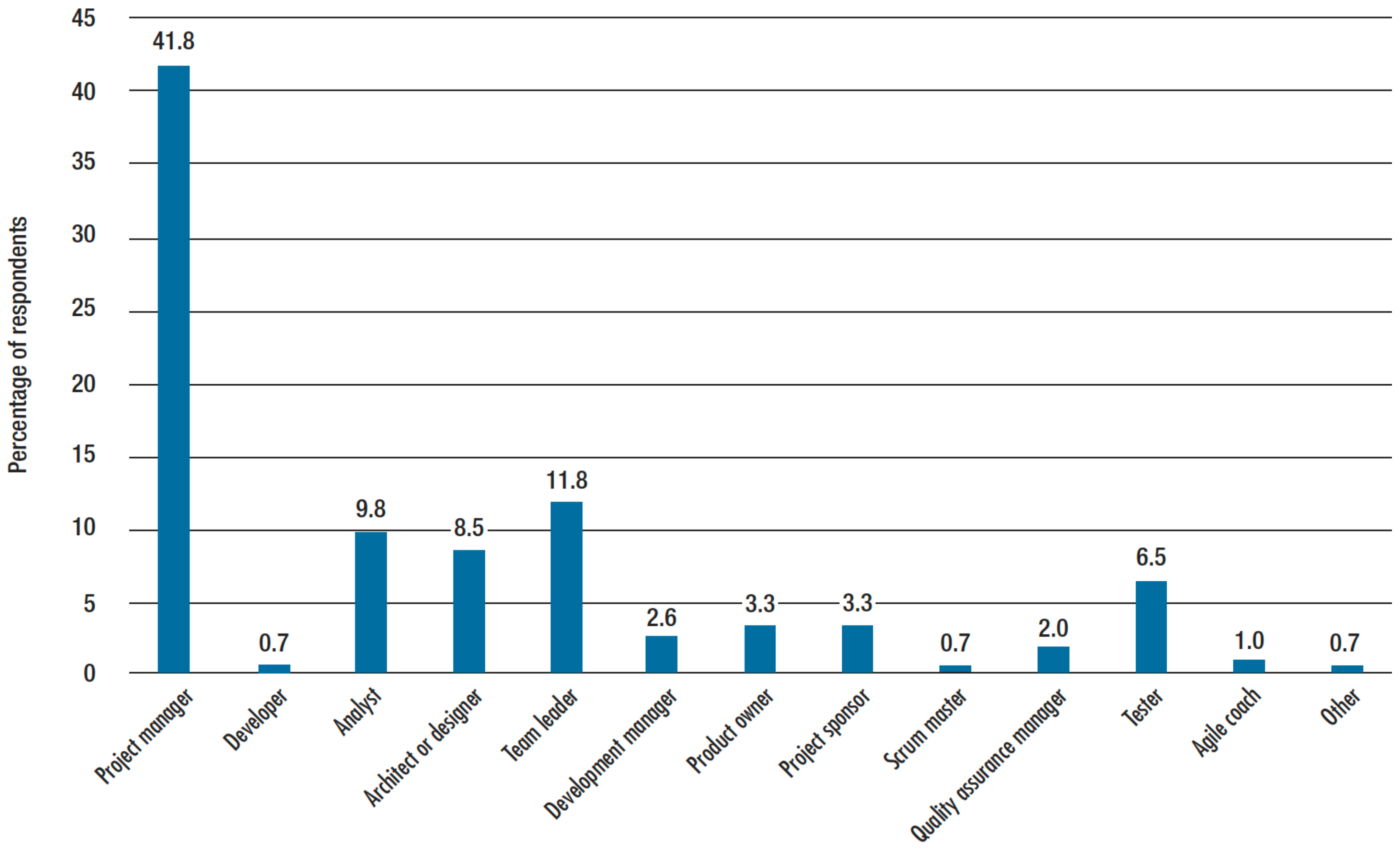
Goals:

(1) Empirically assessing the extent to which different software development methodologies, including **traditional**, **iterative**, and **agile**, are in use.

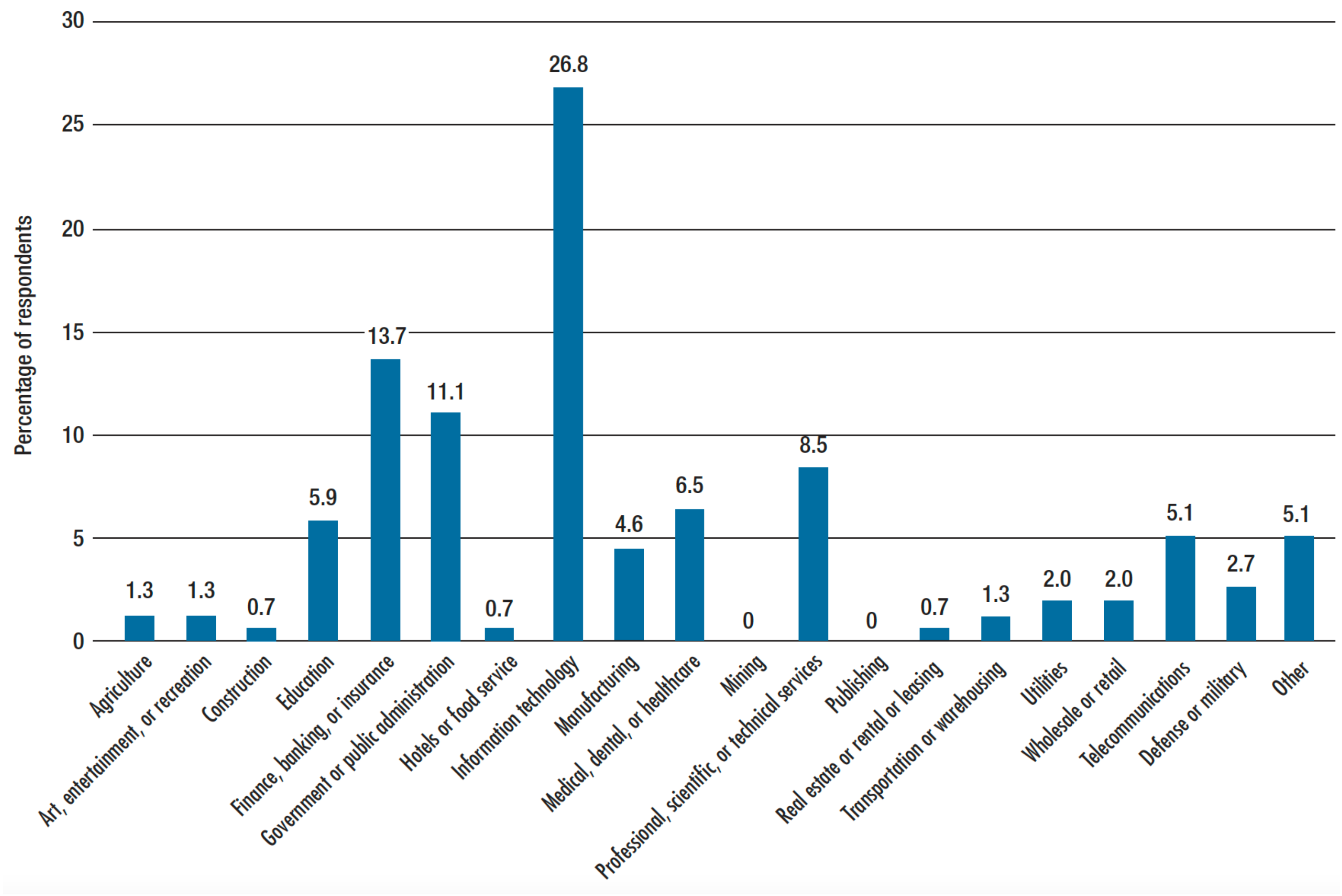
(2) Determining the associations between the adopted methodologies and **organizational**, **project**, and **team characteristics**.

*Vijayasarathy, Leo R., and Charles W. Butler. "Choice of software development methodologies: Do organizational, project, and team characteristics matter?." IEEE software 33.5 (2016): 86-94.

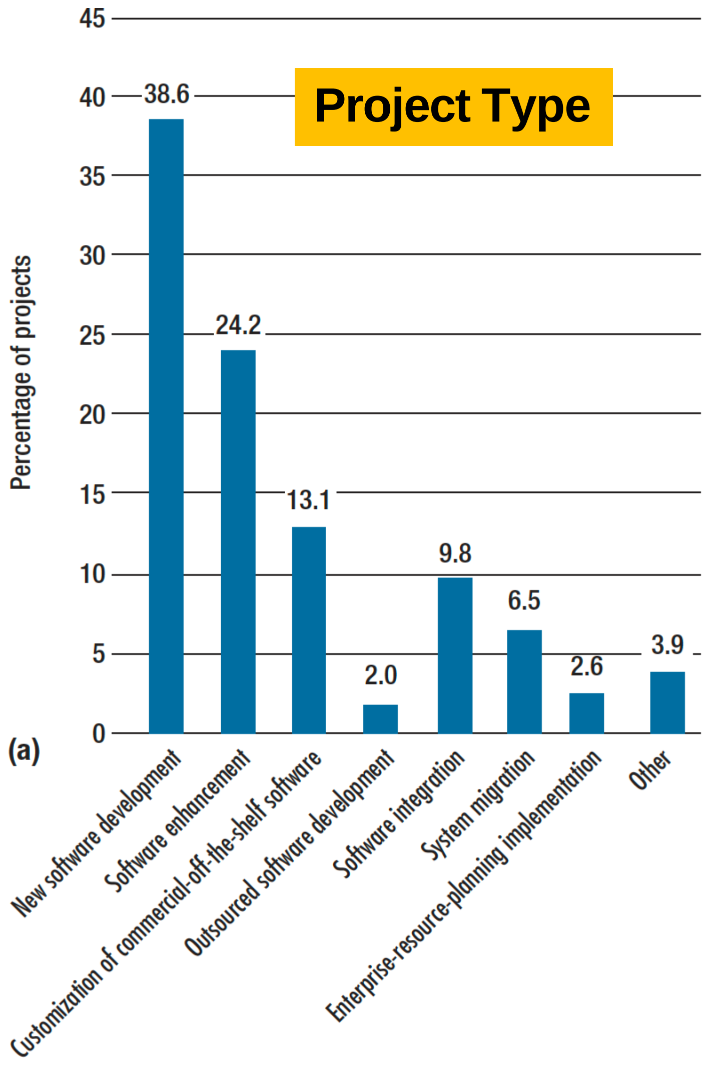
The respondents' primary project roles
(<u>153 Respondents</u>)

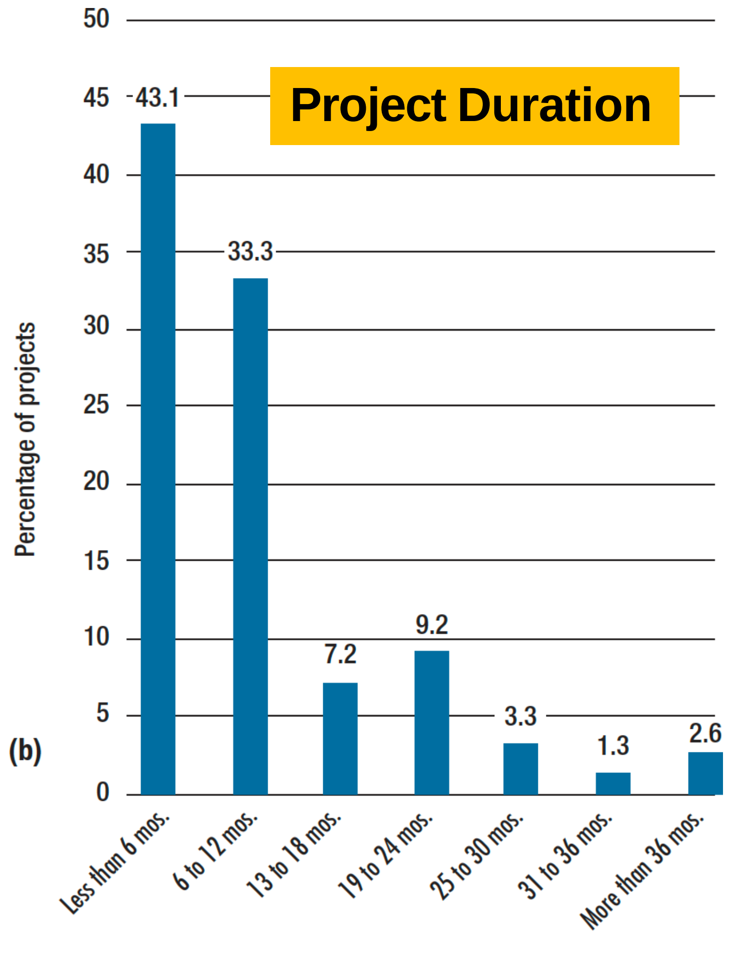


Industry Sector/Domain

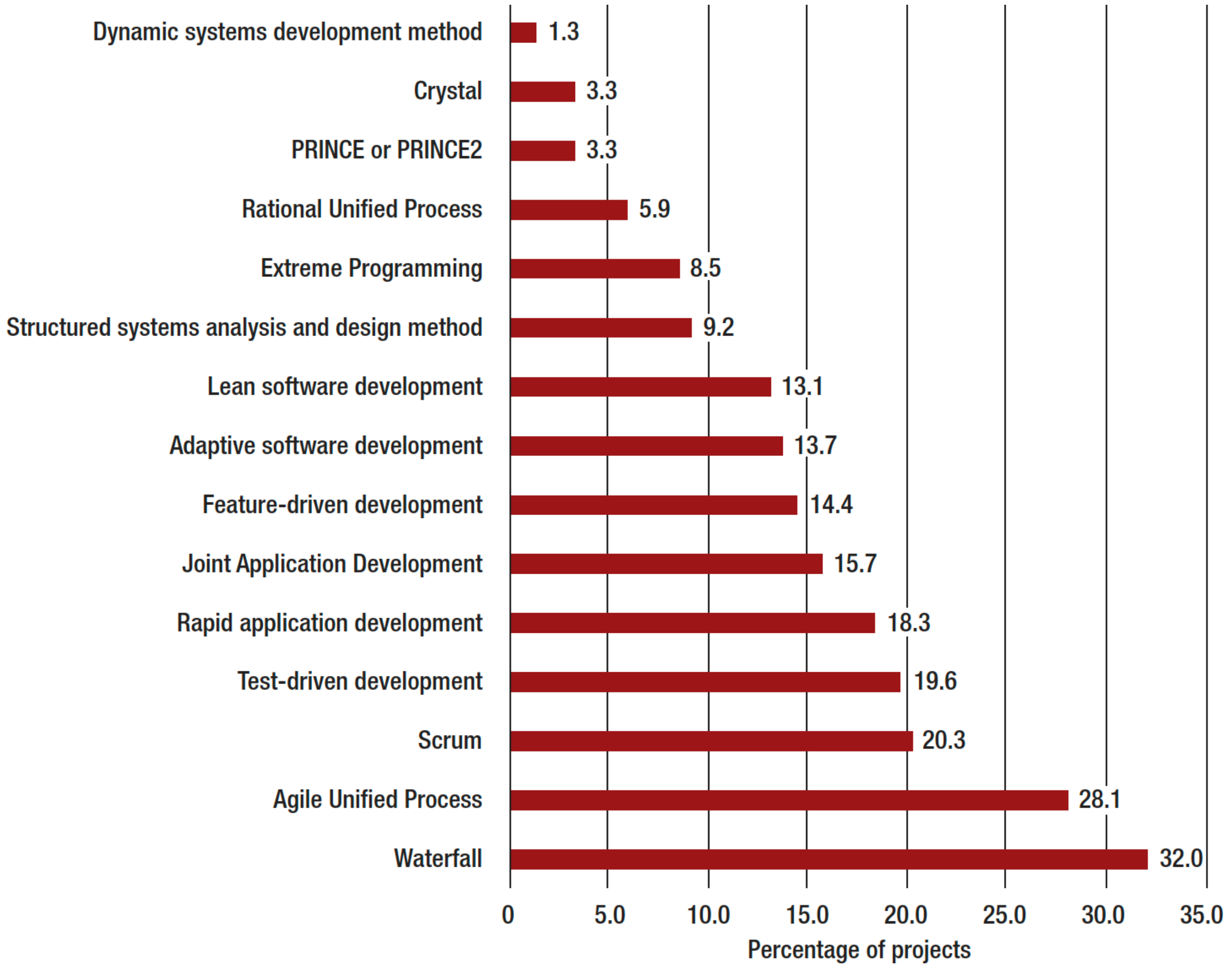


Project Profile

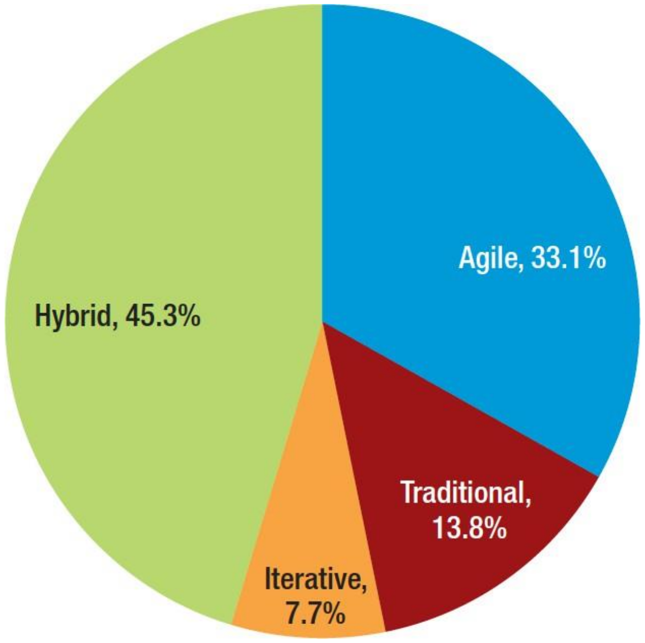




Software Development Methodologies used!



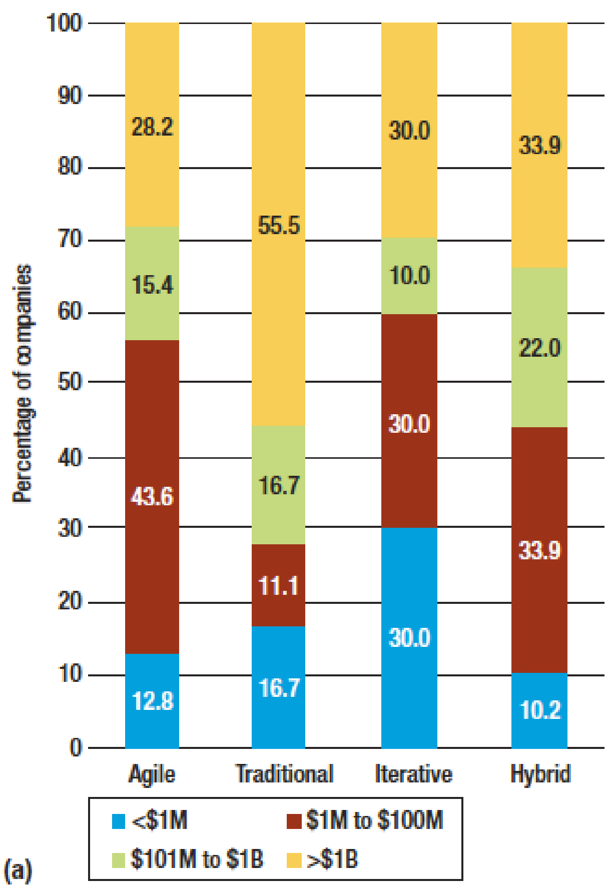
Projects frequently used multiple methodologies

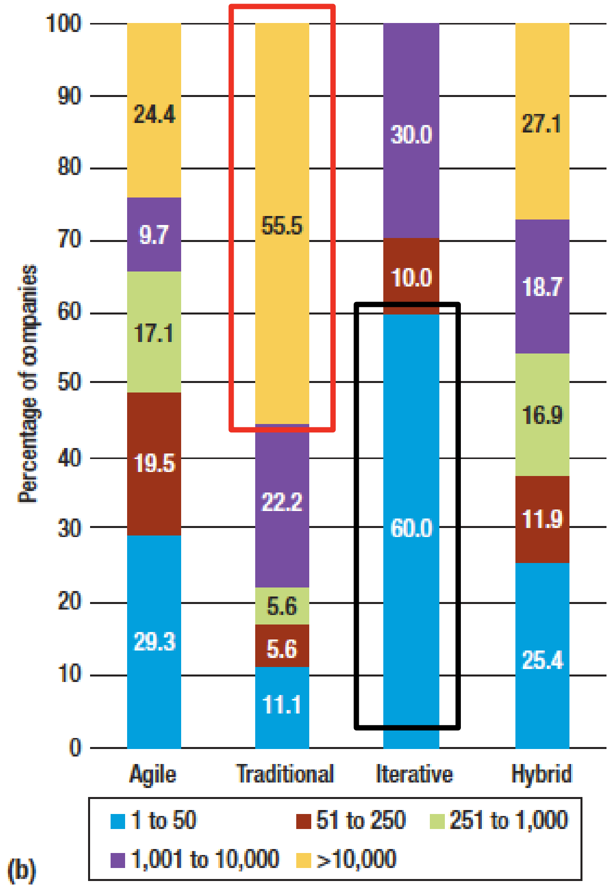


- The **traditional approach** segment includes projects that adopted one or more plan-driven, sequential methodologies such as the **waterfall model**.
- The **agile** segment includes projects that used Agile Unified Process, Scrum, test-driven development, feature-driven development, adaptive software development, lean software development, Extreme Programming, Crystal, and dynamic systems development.
- The **iterative** segment includes projects that used Rational Unified Process, Joint Application Development, and rapid application development.
- The **hybrid** segment includes projects that blended methodologies from the other segments.

Organizational Factors: (a) annual revenue (US$) and (b) number of employees

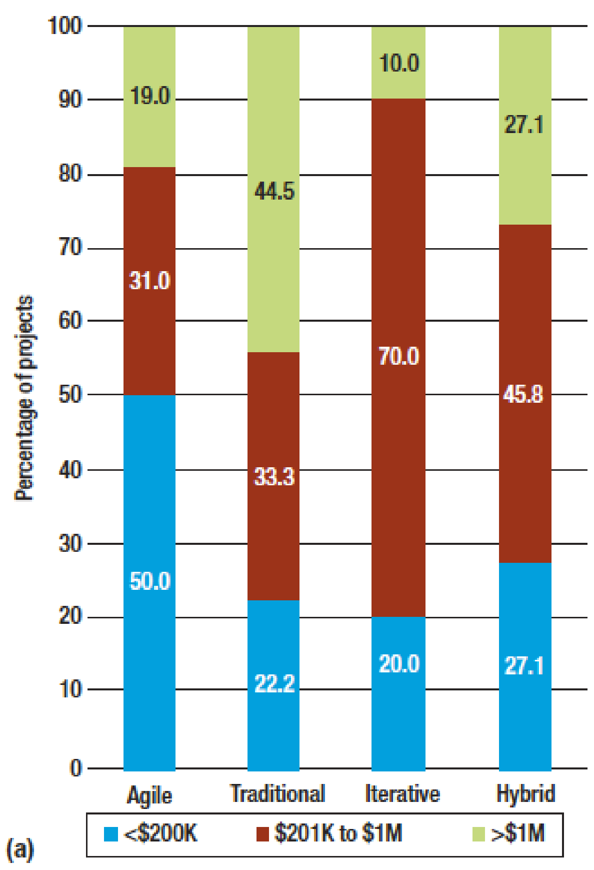
The annual revenue did not have a significant association with the chosen approach but the number of employees had.

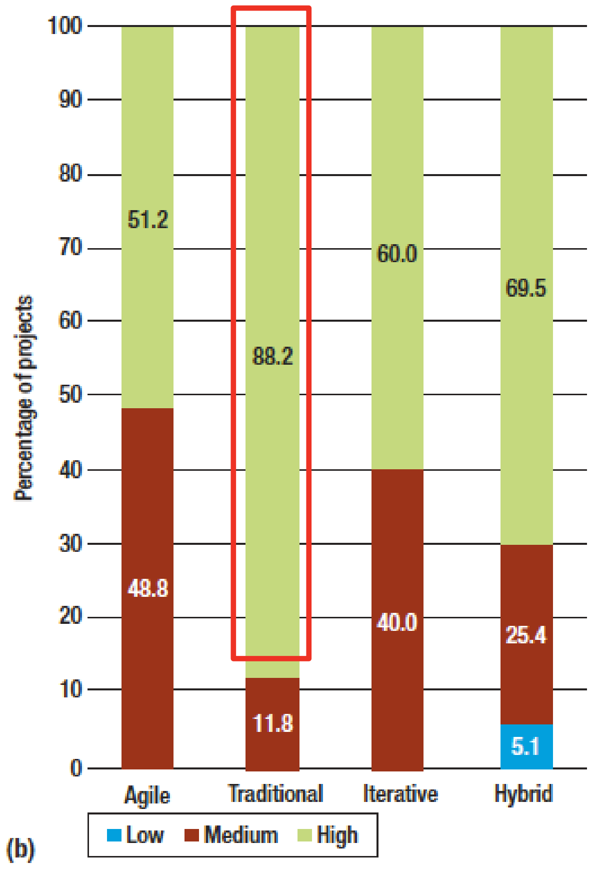




Project Factors: (a) the budget and (b) project criticality

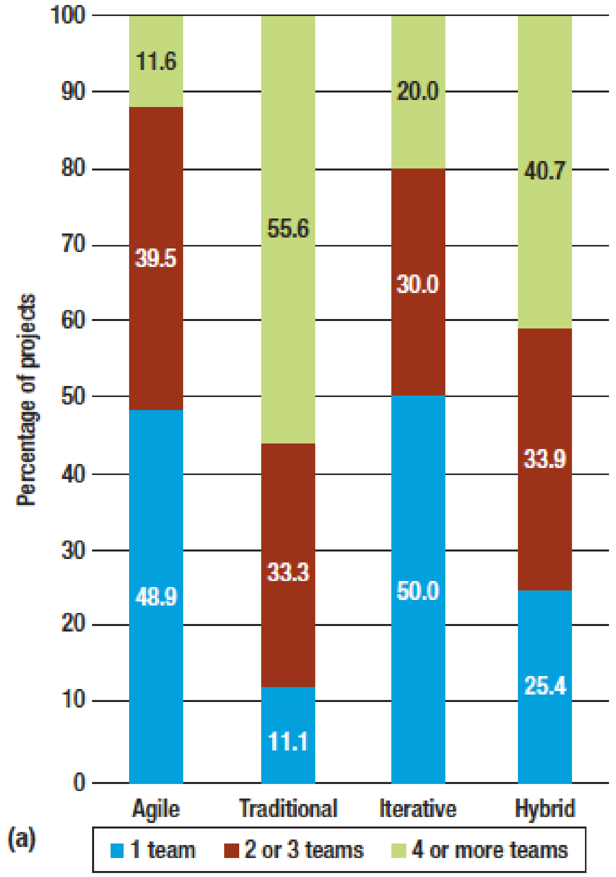
Both factors had a significant association with the development approach

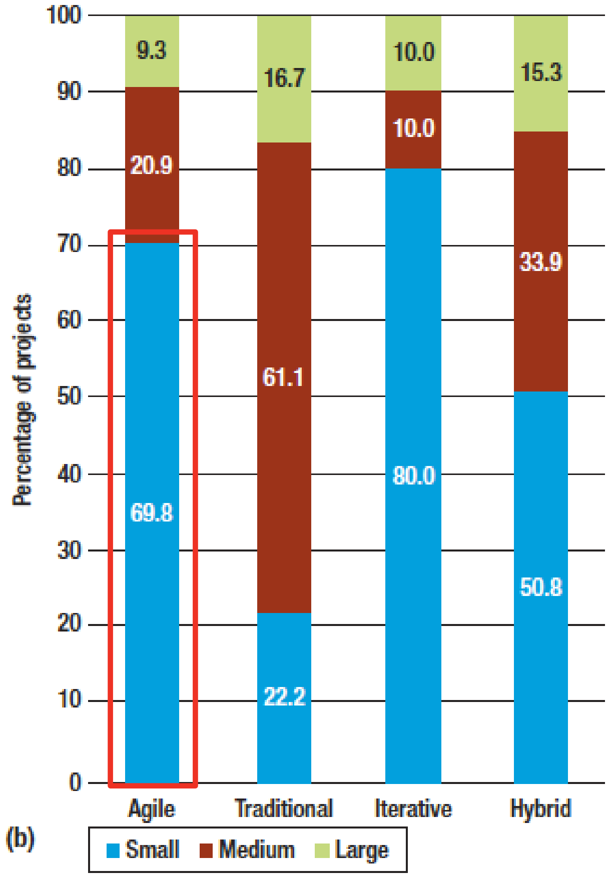




Team Factors: (a) the number of teams and (b) team size

The number of teams and team size had a significant relationship with the development approach.





Small (<10)
Medium (11 to 30)
Large (>30)

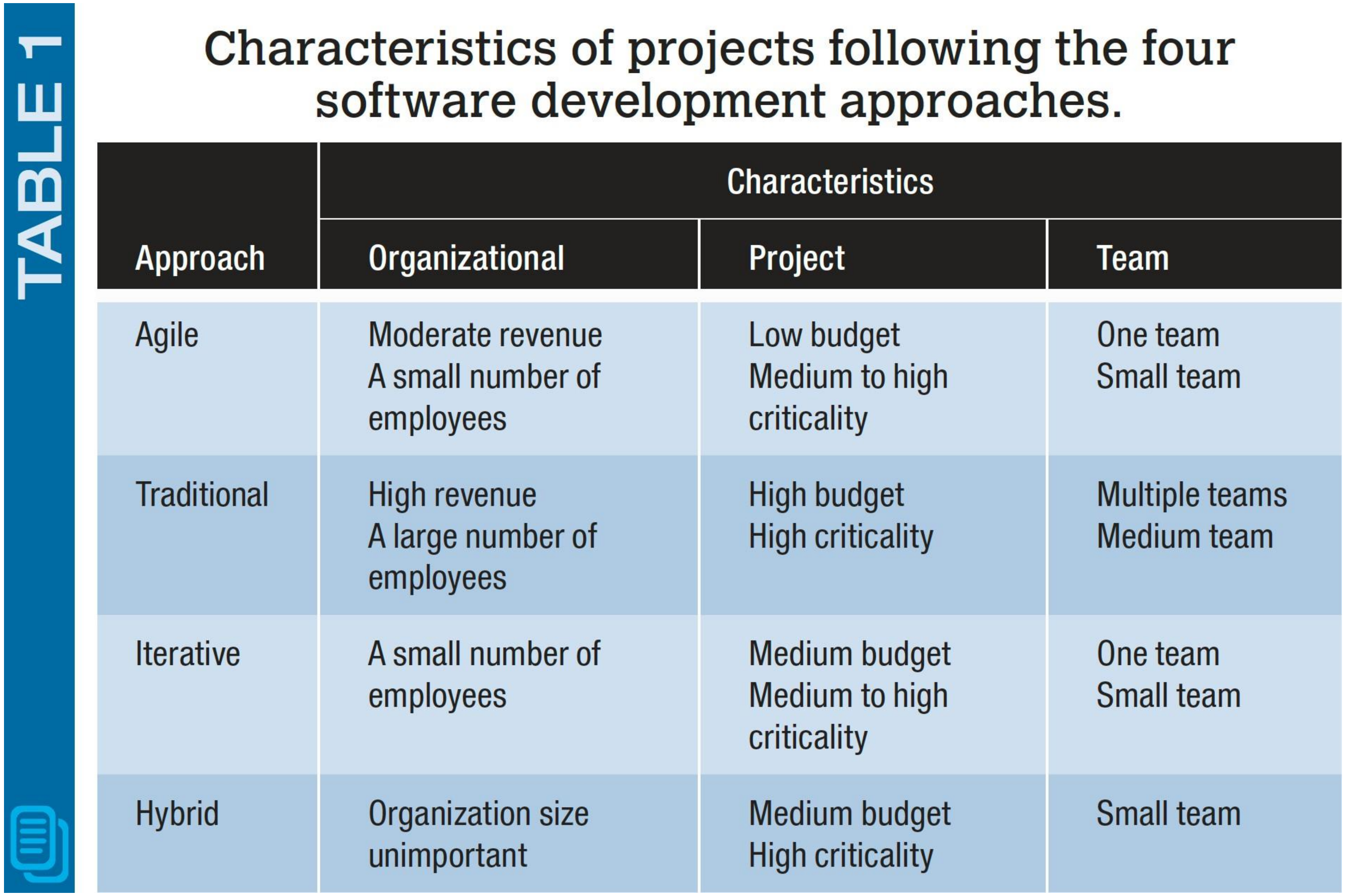
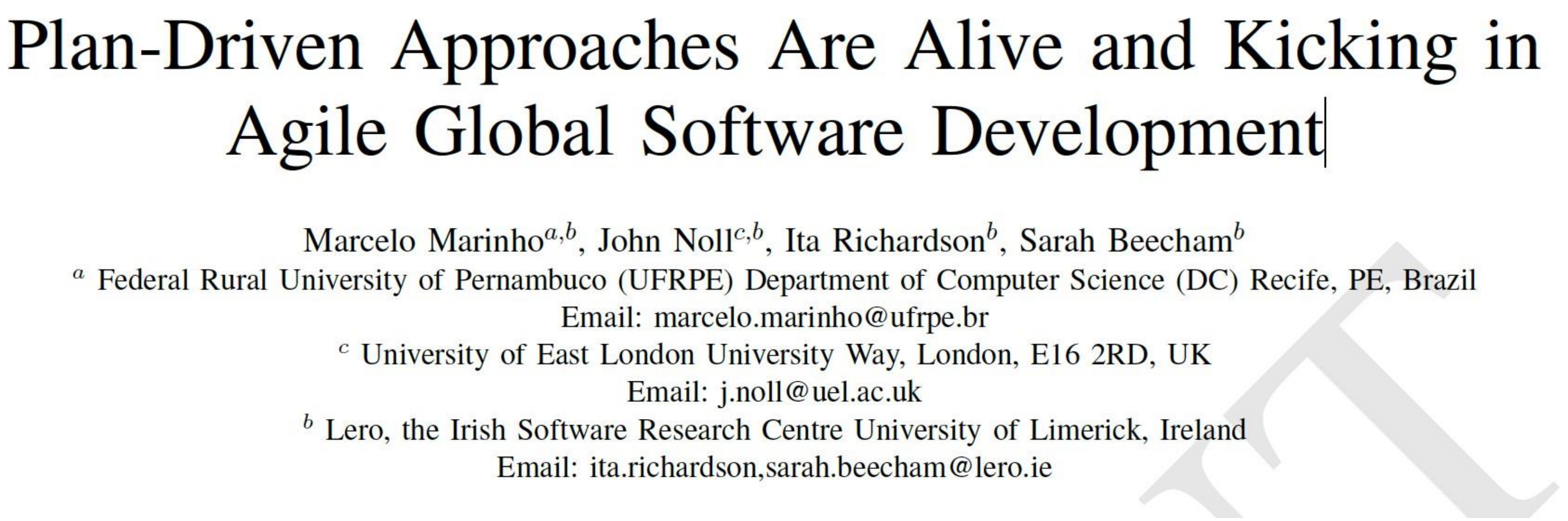


TABLE 1

Characteristics of projects following the four software development approaches.

Approach	Characteristics		
	Organizational	Project	Team
Agile	Moderate revenue A small number of employees	Low budget Medium to high criticality	One team Small team
Traditional	High revenue A large number of employees	High budget High criticality	Multiple teams Medium team
Iterative	A small number of employees	Medium budget Medium to high criticality	One team Small team
Hybrid	Organization size unimportant	Medium budget High criticality	Small team



Plan-Driven Approaches Are Alive and Kicking in Agile Global Software Development

Marcelo Marinho[a,b], John Noll[c,b], Ita Richardson[b], Sarah Beecham[b]
[a] Federal Rural University of Pernambuco (UFRPE) Department of Computer Science (DC) Recife, PE, Brazil
Email: marcelo.marinho@ufrpe.br
[c] University of East London University Way, London, E16 2RD, UK
Email: j.noll@uel.ac.uk
[b] Lero, the Irish Software Research Centre University of Limerick, Ireland
Email: ita.richardson,sarah.beecham@lero.ie

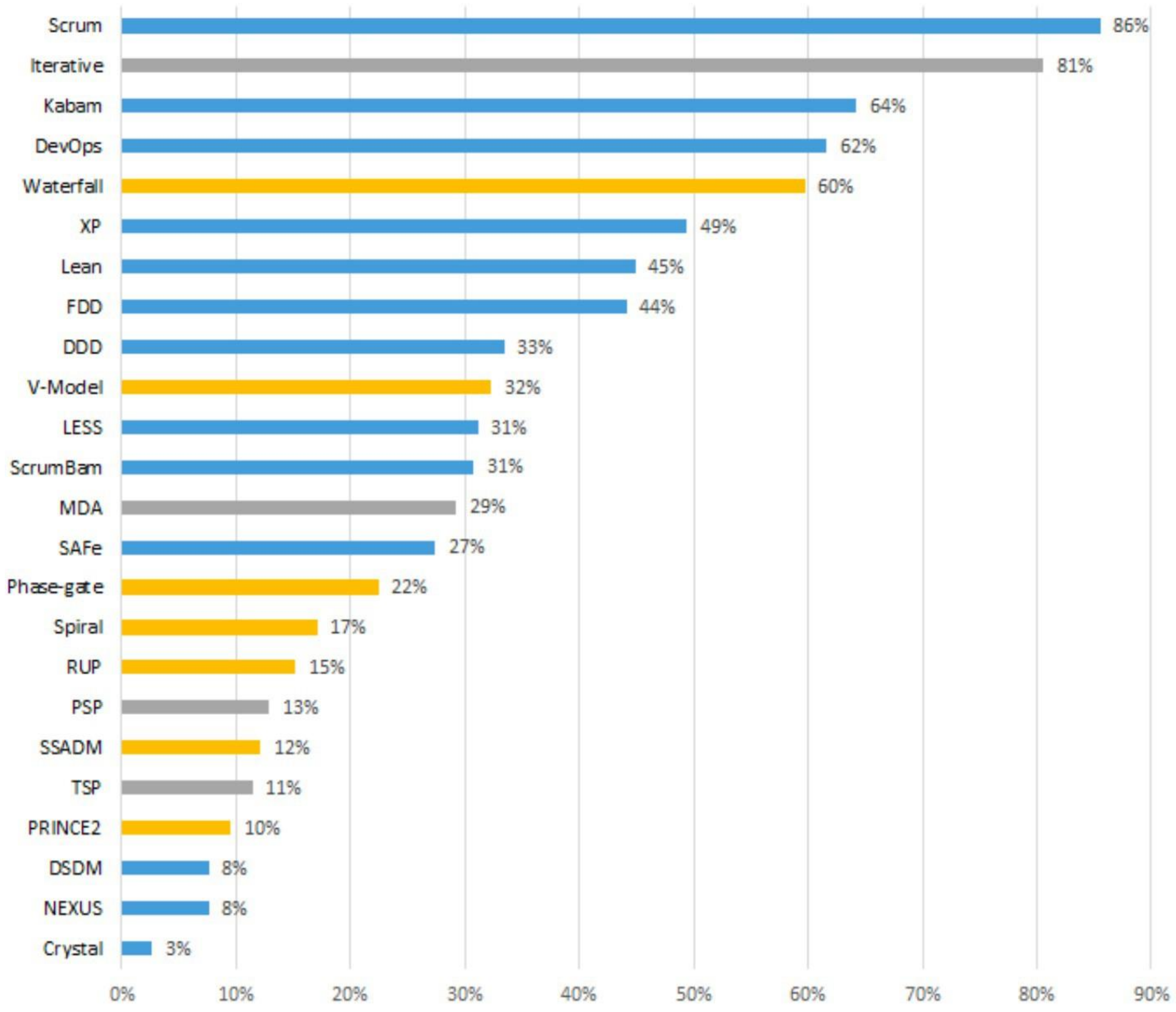
A Research Study in 2019*

Goals:

Gain a deeper understanding of how organizations adopt development methods when developing globally distributed software projects.

*M. Marinho, J. Noll, I. Richardson, and S. Beecham: Plan-Driven approaches are alive and kicking in agile Global Software Development. Empirical Software Engineering and Measurement (ESEM), 2019

Framework usage frequency (n=263). colours: agile: blue, traditional: orange, generic: grey.



Question

Iterative and incremental software development could be very effectively used for customers who do not have a clear idea about the systems needed for their operations. **Discuss the reasons!**

Question

Explain why software testing should always be an incremental, staged activity. Are programmers the best people to test the programs that they have developed? **Discuss the reasons!**

Next Week

Lecture/Lectorial

- Requirements Engineering
  - Types of requirements
  - Techniques to collect, analyze and document requirements

Practical

- Software Process, Unified Process, and Visual Paradigm Tool

References

- Ian Sommerville, Software Engineering, 10th Edition, 2015.
- Roger Pressman, Software Engineering: A Practitioner's Approach, 8/e (McGraw-Hill 2014). 2014 by Roger Pressman.
- Vijayasarathy, Leo R., and Charles W. Butler. "Choice of software development methodologies: Do organizational, project, and team characteristics matter?." IEEE software 33.5 (2015): 86-94.
- Baetjer, Jr., H., Software as Capital , IEEE Computer Society Press, 1998, p. 85.

Many Thanks

Dr Halil Ali

halil.ali@rmit.edu.au