



WHERE INDUSTRY MEETS ACADEMIA: AN ANALYSIS OF SOFTWARE DEVELOPMENT METHODOLOGIES IN PRACTICE
Keenan Robert Ryder (Faculty Mentor: Brad Williams)
Department of Entrepreneurship and Strategy

The landscape of management in a software company is constantly shifting. Not only are new frameworks, languages and technologies released constantly which change the tech stacks, skill sets and methods which are used to produce products, but considerations such as the company’s position along the Business Lifecycle, the size of the development team, the type of project, and the competency of the engineers, result in an endless need for managers to adapt. Perhaps the key management consideration for a technology company is what software development methodology will be employed.

Popular software development methodologies generally fall into three categories: Waterfall, Agile and Spiral. The focus of this thesis was to examine the three main software development methodologies in academic thought and industry implementation utilizing published literature and first-hand interviews with industry professionals. These interviews identified the pros and cons of the “academic” version of each methodology and explored the variety of factors that influence how each methodology is successfully implemented in the real world as well as the impact of the methodologies on various aspects of their businesses.

The clarity of customer needs^a, cost of management^b, required training^c, amount of documentation^d, flexibility in iteration^e and release schedule^f are all key factors to consider in the selection and application of a software development methodology. Based on a review and analysis of published literature, the necessity of each factor for the successful use of each of the major methodologies was ranked (Table 1).

Table 1. Necessity of Key Factors for Successful Use (Low, Medium, High)

FACTOR	AGILE	WATERFALL	SPIRAL
Information^a	Low-Medium	High	Medium
Management^b	High	Low	High
Training^c	High	Low	High
Documentation^d	Low	High	High
Iteration^e	High	NA	Medium
Release^f	High	Medium	High

As indicated in the table above, Agile and Spiral are both more flexible than Waterfall regarding schedule of iteration and release, and clarity of software requirements at the outset of a project. Agile requires less documentation overall than either Waterfall or Spiral. However, both Agile and Spiral require managers to have special experience and developers to have special training, making both methodologies more difficult to implement than Waterfall.

To evaluate the applicability of these findings to the local tech environment and to expand the body of knowledge on this subject, three companies from the Greater Salt Lake City area (Pluralsight, Zibtek and Tafi), each at a different stage in the Business Lifecycle, were interviewed. The goal was to better understand how and why each company selected and utilizes one (or more) of these software development methodologies, how they may have been adapted it to suit their specific business or project needs, and how their preferred methodology interacts with their development costs, product release schedule, sales cycle, marketing, organizational behavior and reporting structures, role of middle and upper management, and customer satisfaction.

Table 2. Usage of Software Methodologies by Surveyed Companies

Companies Interviewed (Stage in the Business Lifecycle)	AGILE	WATERFALL	SPIRAL
Pluralsight (Expansion/Maturity)	X		
Zibtek (Growth & Establishment)	X	X	
Tafi (Startup)	X		X

Key findings and conclusions from the analysis of published literature and the industry interviews are summarized in the recommendations below:

- Each software development methodology should be viewed as a philosophy and a tool set which can be implemented selectively to suit the project needs and skills/experience of the development team.
- It is more productive to take the above selective approach than it is to adhere rigidly to a single methodology.
- Customers appreciate firms that use a methodology that is flexible and promotes frequent releases, e.g. Agile, Spiral. However, methodologies that promote frequent releases necessitate more upfront planning and requirement definition.

- It is important that engineers understand the business context of software features before implementation.
- Large projects undertaken with Agile should be divided between multiple small teams each with a distinct focus.
- There are very few instances where Waterfall will be preferred over Agile. This is largely due to the flexibility needed in modern software development with its high level of customization.
- Spiral is the best methodology for high risk and loosely defined projects but an acceptable level of risk and acceptable end product must be defined to avoid infinite cycles.
- A small project (such as an internal tool) will only be slowed down by using a formal software development methodology. A trimmed-down version of Waterfall may be used to provide documentation should future developers need to revisit the project.
- The most important factors to consider when choosing a software development methodology will always be the clarity or ambiguity of the project, the scope of the project (size and level of customization), and the skills/experience of the development team.
- As a company moves through the Business Lifecycle (Seed & Development > Startup > Growth & Establishment > Expansion > Maturity), the corollary movement through categories of software development methodologies may be:
None/Spiral elements > Spiral/Agile > Agile (Well-defined projects)/Spiral (Loosely-defined/High-risk projects)/ Waterfall elements (Small/Well-defined projects)

These recommendations provide a perspective on the use of the three software development methodologies grounded in industry practices and supported by academic research. These may be particularly useful for new managers and entrepreneurs in the software development industry.