



---

**RESEARCH REVIEW**

# Assumptions Underlying Agile Software-Development Processes

Daniel Turk, Colorado State University, USA  
Robert France, Colorado State University, USA  
Bernhard Rumpe, Braunschweig University of Technology, Germany

---

## ABSTRACT

*Agile processes focus on the early facilitation and fast production of working code, and are based on software-development process models that support iterative, incremental development of software. Although agile methods have existed for a number of years now, answers to questions concerning the suitability of agile processes to particular software-development environments are still often based on anecdotal accounts of experiences. An appreciation of the (often unstated) assumptions underlying agile processes can lead to a better understanding of the applicability of agile processes to particular situations. Agile processes are less likely to be applicable in situations in which core assumptions do not hold. This article examines the principles and advocated practices of agile processes to identify underlying assumptions. It also identifies limitations that may arise from these assumptions and outlines how the limitations can be addressed by incorporating other software-development techniques and practices into agile development environments.*

*Keywords:* agile development; assumptions; extreme programming; limitations

---

## INTRODUCTION

As more organizations seek to gain competitive advantage through the timely deployment of services and products that meet and exceed customer needs and expectations, developers are under increasing pressure to develop new or enhanced implementations quickly (Cusumano & Yoffie, 1999). While iterative processes have been helpful in developing, modifying, and maintaining systems more quickly

and more successfully (Basili & Turner, 1975; Boehm, 1986), agile software-development processes have been evolved primarily to support the timely and economical development of high-quality software that meets customer needs at the time of delivery. It is claimed by agile-process advocates that this can be accomplished by using development processes that continuously adapt and adjust to (1) the collective experience and skills of the developers, including experience and skills gained thus far in the de-

velopment project, (2) changes in software requirements, and (3) changes in the development and targeted operating environments. Examples of published agile processes are extreme programming (XP; Auer & Miller, 2002; Beck, 2000; Beck & Fowler, 2001; Jeffries, Anderson, & Hendrickson, 2001; Newkirk & Martin, 2001; Wake, 2002), the Crystal process family (Cockburn, 2001), SCRUM (Rising & Janoff, 2000; Schwaber & Beedle, 2001), adaptive software development (Highsmith, 2001), and the AUP (agile unified process; Larman, 2001), which has grown out of work on the UML (unified modeling language; Jacobson, Booch, & Rumbaugh, 1999; Object Management Group [OMG], 2004b; Rational Corporation, 1998).

The proper use of agile processes requires an understanding of the situations in which agile processes are and are not applicable. One way of determining whether an agile process is applicable in a particular situation is to check whether the assumptions underlying the process hold in that situation. If the assumptions do not hold, then the use of the agile process may not be appropriate. Prevailing descriptions of agile processes seldom present the underlying assumptions explicitly, and thus it is difficult for developers and project planners to determine the applicability of agile processes to their projects and environments.

This article identifies some of the assumptions underlying agile processes that can be used to help determine the applicability of agile processes to particular situations. The article also discusses some of the limitations that may be inherent in agile approaches because of these assumptions. The assumptions were identified by examining published work on XP (Auer & Miller, 2002; Basili, Caldiera, & Rombach, 1994), SCRUM (Schwaber & Beedle, 2001), the AUP as described by Craig Larman (2001), agile modeling (Ambler, 2004), critiques of agile processes (Boehm, 2002; McBreen, 2003), and the principles stated by the Agile Alliance (2004).

It is important to note that our critique of agile processes is concerned with identifying assumptions underlying a family of agile processes. Other critiques (Boehm, 2002; Boger, Baier, Wienberg, & Lamersdorf, 2001; McBreen,

2003) and assessments (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003; Boehm & Turner, 2004; Glass, 2001; Kontio, Høglund, Ryden, & Abrahamsson, 2004; Sfetos et al., 2004) of agile processes have been published, but none of the ones we have examined have focused on identifying assumptions underlying agile processes for the purpose of determining the scope of their applicability. For example, in the book *Questioning Extreme Programming*, McBreen presents a critique of XP in which he poses some important open questions and provides answers to other questions based on his personal experience, but he does not explicitly identify assumptions underlying agile processes. McBreen's critique was used as a source in our work along with other experiences reported elsewhere (Crocker, 2001; Lindvall et al., 2002).

The remainder of the article is structured as follows. We give an overview of a typical agile process, extreme programming, to give the reader a concrete example of an agile process. Then we describe the assumptions that we have identified. Next we identify some of the limitations that arise in situations in which these assumptions are not met and suggest how they can be addressed by adapting some of the agile-process techniques and practices. We conclude with an overview of the results of our work and an outline of issues that require further investigation.

## **OVERVIEW OF EXTREME PROGRAMMING: A REPRESENTATIVE AGILE PROCESS**

There are a variety of software-development processes that currently claim to be agile. Space does not allow us to give an overview of all of the agile processes we have reviewed. However, since XP is probably the most well-known agile process (Beck, 2000; Strigel, 2001), we use it to illustrate representative agile-process concepts.

### **Extreme Programming (XP)**

It can be argued that the popularity of XP helped pave the way for other agile pro-

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/article/assumptions-underlying-agile-software-development/3342](http://www.igi-global.com/article/assumptions-underlying-agile-software-development/3342)

## Related Content

---

### A Machine Learning and Large Language Model-Integrated Approach to Research Project Evaluation

Jian Ma, Zhimin Zheng, Peihu Zhu and Zhaobin Liu (2024). *Journal of Database Management* (pp. 1-14).

[www.irma-international.org/article/a-machine-learning-and-large-language-model-integrated-approach-to-research-project-evaluation/345400](http://www.irma-international.org/article/a-machine-learning-and-large-language-model-integrated-approach-to-research-project-evaluation/345400)

### A Run-Time Based Technique to Optimize Queries in Distributed Internet Databases

Latifur Khan, Arunkumar Ponnusamy, Dennis McLeod and Cyrus Shahabi (2003). *Advanced Topics in Database Research, Volume 2* (pp. 128-161).

[www.irma-international.org/chapter/run-time-based-technique-optimize/4344](http://www.irma-international.org/chapter/run-time-based-technique-optimize/4344)

### Mapping Fuzzy EER Model Concepts to Relations

Jose Galindo, Angelica Urrutia and Mario Piattini (2006). *Fuzzy Databases: Modeling, Design and Implementation* (pp. 171-178).

[www.irma-international.org/chapter/mapping-fuzzy-eer-model-concepts/18763](http://www.irma-international.org/chapter/mapping-fuzzy-eer-model-concepts/18763)

### Privacy-Preserving Contact Tracing for Curbing the Spread of Infectious Disease

Hui Li and Yifei Zhu (2023). *Journal of Database Management* (pp. 1-17).

[www.irma-international.org/article/privacy-preserving-contact-tracing-for-curbing-the-spread-of-infectious-disease/324075](http://www.irma-international.org/article/privacy-preserving-contact-tracing-for-curbing-the-spread-of-infectious-disease/324075)

### NoSQL Database Phenomenon

(2018). *Bridging Relational and NoSQL Databases* (pp. 34-93).

[www.irma-international.org/chapter/nosql-database-phenomenon/191980](http://www.irma-international.org/chapter/nosql-database-phenomenon/191980)