This work will be presented at WSC 2018 Conference
December 9-12, Gothenburg, Sweden
As part of an invited presentation

Beyond SAFe® Framework

Telemetry Driven and Simulation Augmented Agile Development

Joseph Barjis
/PhD, SAFe® Framework Contributor/

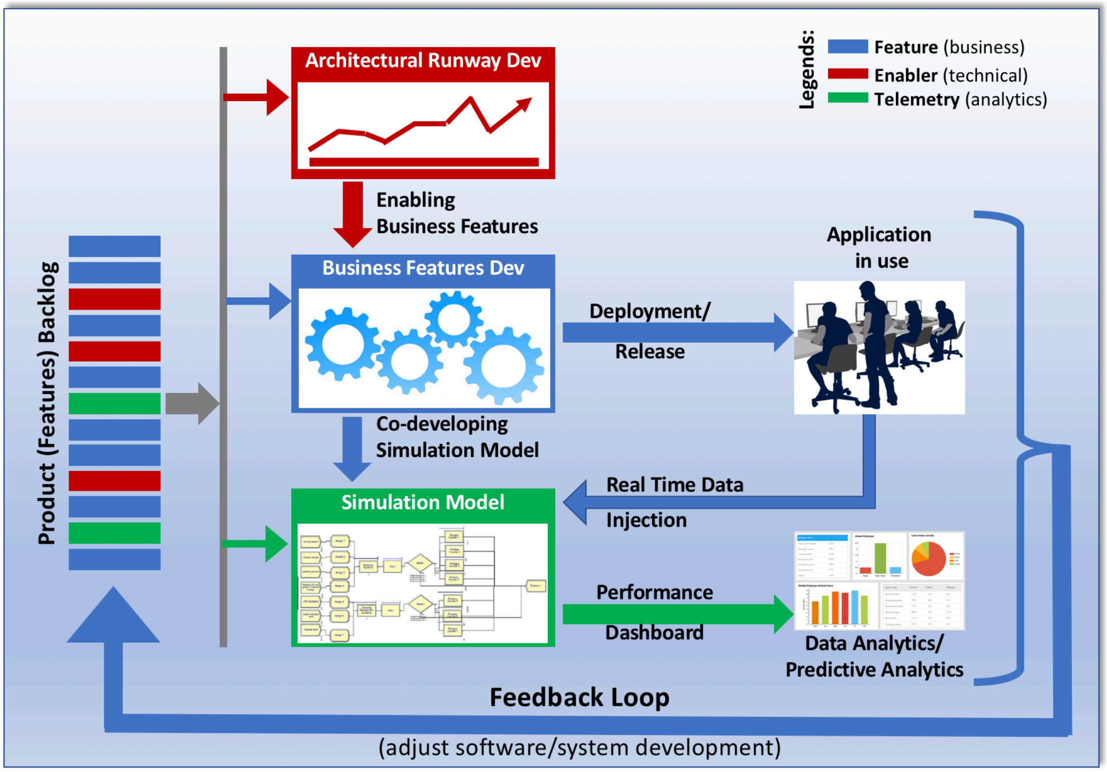

Figure 1: SAFe® Framework extended with data analytics and simulation

Introduction

In the last two decades, nothing shaped and transformed the practice of software and system development as radically as the Lean-Agile practice. In the core of Agile practice are iterative development and continuous adjustment of the system scope and functionality based on the feedback of earlier delivered pieces of functionality. To this end, the notion of feedback loop constitutes the very essence of Agile practice.

On the other hand, in an information age that data becomes new natural resources and both application execution and user interaction contribute to the pool of big data, automatic data collection must be integral to the development of new applications. Data collection should be an embedded capability of each software feature on top of the business function that they support. So, they not only deliver business functionality but also collect data on their own performance. As natural resources, raw data present less value if they are not accessible, usable, and customized for the user needs.

In addition to these two premises, the importance of data analytics and predictive analytics is ever growing in new system development as well as existing systems. Complex decisions require tools that drive on data analytics.

Beyond SAFe® Framework

In the Agile system and software development practice, new applications are developed by continuously adding new functionalities, usually referred to as business Features (see Figure 1, dark blue color items in the product backlog) and improving the application architecture and environment, usually referred to as Enablers (see Figure 1, dark red items). Next to that is continuous deployment, including in the production environment, which is accomplished when development and operation teams work for the same goal, fast value delivery to customers. This collaboration is coined as DevOps. This is where automatic data collection is important to get insight into the behavior of each change introduced through a new Feature to the existing application, and to see whether the change impacts the performance of the system or application as a whole. Furthermore, these data help to see the business use of a Feature and whether the benefit hypothesis for that Feature holds true.

To weave data analytics capability right into the fabric of new applications, Agile practice, in addition to Features and Enablers, should encourage to add data analytics items in the product backlog (Figure 1, see green color items). These data analytics backlog items will comprise building blocks that a prospective simulation model can be built upon. Such a simulation model is used to predict the application behavior partially based on real data injected from the application execution and partially from mock or archival data to fill in for the Features that are not yet developed or deployed.

In early phases of application development, simulation models can be used to represent not yet existing Features or system components. While the features deployed in production will show an actual behavior of the application so far developed, the simulation model will animate and predict the whole behavior of the application once it is fully developed and the remaining features are implemented. These applications can be fairly complex, expensive, multiyear development, and cross boundaries of many functions and departments.

In essence, here the simulation model plays a multi-purpose role, including filing in the gaps of yet incomplete features. Also, simulation outcomes contribute towards measurement, which is critical in Agile software/system development. This way, measurement and metrics supported and enhanced by the simulation results can be cost effective on one hand and allowing to see the results of otherwise a very lengthily data collection process that takes days or longer, on the other hand.

From Agile software and system development perspective, most important use of embedded simulation model is creation of feedback loop (see Figure 1), which evolves as the application development is further and further advanced by adding new features. This novel application of simulation based on real data and on top of work-in-progress may suggest where to improve performance, which future features are in most demand, and whether users prefer certain ways over others.

Furthermore, the intentional introduction of chaos to study robustness and resilience of the application can be first studied at the simulation model level and then, in a controlled manner, introduced into the production environment. Agile system development encourages Set-Based Design, which entails to test variety of design options. A simulation model that as input receives real time data through dynamic data injection can more accurately establish which design option affords most optimal solution.

Challenges

With the potential that use of simulation yields when building simulation model on top of applications that are work-in-progress comes a few challenges. First, the *maintainability* of the simulation model due to continuous deployment of new features and need for more telemetry data. Second, *visualization* or data presentation, i.e., building a dashboard that present the data in an easy fashion and making sense to business and executives, as well as to developers and analyst. These challenges present a great opportunity for profound new research.

The simulation model maintainability challenge is dominantly technical challenge, which needs to be addressed through new simulation frameworks and simulation processes. It is similar to the issue of software evolution and keeping documentation updated. Here, one could borrow concepts from *Behavior-Driven Developm*ent, where description of the behavior that a software application has to implement is interwoven into the software itself. That is, a new functionality cannot be developed unless a new behavior is specified and documented, and vice versa, a software development is not complete unless it passes the specified behavior. Such a loop constantly keeps each other, i.e., documentation and the software itself, in correspondence, and, thus, always updated.

The situation is a bit different with addressing visualization. Visualizing the collected data from the application execution in production environment entails a great deal of data engineering, data science, and data manipulation. Furthermore, this also fosters close collaboration of technical and business people. More specifically, collaboration of developers, testers, and business analysts. This collaboration is also known as three amigos, which emphasizes the importance of different perspectives on the system to be built. While nominally they are referred to as three amigos, in fact, more perspectives are needed. One such important perspective is of the user experience or UX design.

In this regard, simulation elevates to the mainstream of Agile system and software development by assuming the role of collaboration and communication tool as well. In this role, simulation is not merely a technical model or predictive tool, but a vehicle to collaborate, understand, and adjust the system development process.

Finally, there is input data challenge through dynamic data injection. Data injection challenges are coming from a myriad of factors: the production data is just emerging. In each given moment, the simulation is run on richer data set than the previous runs. The data injected onto simulation model from the application is not complete as many of the features are not yet developed. Thus, any optimization of the application will be challenged when new features are implemented in production. Balancing between the data injected from the production environment, real time data, and mock data need to be addressed too.

Author:

Dr. Barjis has 25+ years of technology and business acumen in consulting, R&D, training, and teaching.

He is a thought leader and expert in Agile, Systems Engineering, Enterprise Engineering, Organization Design, Software Development, Information Systems Design, Predictive Analytics, Modeling and Simulation. He has been internationally invited for talks, scientific seminars, or lectures. He worked from top research universities to top consulting companies.

Dr. Barjis research works are published in 100+ peer refereed outlets (over 25 journal articles and editorials; 17 book chapters; 18 edited proceedings books, 10 Special Issues Editor; 60+ papers in peer refereed conference; published in high impact journals), including a number of Best Paper Awards. He is co-founder of the emerging discipline of Enterprise Engineering and founder of an international workshop in the field, now on its 14th annual edition.