A Day-in-the Life of a Systems/Software Engineer 3-5 Years Ahead

Maria Ericsson [a] and Sky Matthews [b]

[a] *Software Group Rational Europe*
IBM,
Stockholm, 164 92, Sweden

[b] *CTO Office of Software Group Rational*
IBM
Durham, NC 27703-9141, USA

ABSTRACT

Key technology shifts such as big data, social technologies, internet-of-things, and mobile are reshaping the landscape in which engineers operate. This paper will present a vision of software and systems engineering of the future, specifically considering development of complex large-scale real-time embedded systems. We will discuss the future of software/systems craftsmanship, and how this impacts the day-in-the-life of an engineer. In order to address challenges related to speed, quality, and cost we see organizations responding by driving initiatives in the following areas:

- Agile – deliver more often with focus on business outcome
- Lean – take out non-value adding tasks
- Collaboration – a culture of open information sharing
- Innovation – create room for risk-taking
- DevOps – breaking down organizational silos

We also see development practices evolving to respond to the changing landscape in the areas of architecture, steering/governance, people/teams as well as how to manage development platforms.

Keywords: Systems Engineering, Agile, Lean, DevOps, Innovation

INTRODUCTION

The world of systems engineering has traditionally been one shaped by well-defined and predictable processes and methods to ensure high-quality outcomes. In development organizations consisting of hundreds and sometimes thousands of people this has been a necessity in order to put clearly defined roles and quality toll-gates in place to ensure order in creative chaos. Tools play a key part to manage dependencies and traceability between a complex web of project deliverables as well as automate work-intensive tasks such as testing.

Technology and culture shifts are driving towards a more open and collaborative way of working, where the traditional governance principles can be perceived as standing in the way of making progress. In this paper we explore how the tools infrastructure of the future can help break down silos, not only within the development team, but also between the development team and its suppliers and customers, where some of the largest productivity barriers remain due to governance and supply chain practices that have grown and calcified over decades.

In a sense, systems engineering historically is about creating a development factory where people are imagined to act as cogs and wheels in the machinery that produces complex systems or products.

The systems engineering industry is moving towards a significant culture change driven by external factors – the next generation engineers will expect a different level of empowerment to make decisions based on information and people available through e.g. social media and analytics. The pace of change in the market place is increasing rapidly, and being able to quickly react to change in market needs is essential. Where before a new function could take months or years to make available we are now down to expectations of days and weeks. We are also evolving our understanding of how people are incentivized to excel and how in this context the style of management is evolving (Denning, 2010).

Technology advances are now enabling computer-assisted engineering in ways that were not practical only a few years ago. In many areas of engineering, advanced computer-assisted tools help practitioners analyze vast amounts of data and run sophisticated “what if” analyses to help make technical decisions. However, these techniques have largely not been applied to complex software development projects. Decision-making in large-scale software development is still primarily ruled by trial and error, intuition and “SWAGs”[1]. We strongly believe that the next few years will see the application of Big Data and analytics on large scale systems engineering and software development, and practitioners will begin to trust and even depend on machine-driven analyses to cut through large amounts of data to help them make better decisions. Through this paper we explore the potential impact that these techniques may have on the productivity of individuals and teams when they are seamlessly integrated in to our development environment. Our intention is to stimulate dialog on the anticipated impacts these technologies may have, and to get development organizations to start thinking about and planning for the next wave of change in process and ways of working that will be enabled by such advances.

The key technology shifts considered are:

- Cloud,
- Mobile,
- Big data,
- Analytics

The context in which we have been considering the topic is of that of organizations developing complex large-scale real-time embedded systems with long life expectancy.

RESHAPING THE LANDSCAPE IN WHICH ENGINEERS OPERATE

Industry Direction

In order to change the way organizations operate to be more flexible and faster, the industry is looking to strengthen operations through:

Agility: In essence to deliver more often with a relentless focus on business outcome. There is a very strong community formed, an agile movement considering the formulation of the Agile Manifesto (http://www.agilealliance.org/the-alliance/) as its defining moment in time. Its foundation is in software development, but has grown to be applied across engineering organizations (Denning, 2012).

[1] Software Wild-Assed Guess

Lean: Grown out of manufacturing, lean is focused on taking out non-value adding tasks while preserving value. A set of principles is applied to move waste out of the production line. The term was coined in the 80s, but ideas around waste reduction were applied already in the early days of the industrial revolution (Womack, Jones, Roos 1990). An evolution of this thinking are the principles are the ideas on Lean Startup techniques (Ries, 2011)

Collaboration: The idea of collaborative leadership has been discussed since the 1990, what we now also see with the evolution of social techniques as an evolving culture of more open information sharing.

Innovation: Traditional techniques for managing predictable results and driving down risk has a side effect of discouraging doing things you haven't done before. Organizations have and increasing focus on making sure resources are carved out to focus on innovation, in essence create room for risk-taking.

DevOps: This is in essence about breaking down organizational silos. Originating from issues in a traditional IT organization where development hands off and application to operations. The concept has evolved to cover what it means to instrument a lean and agile organization, creating an environment that help overcome organizational barriers and support transparency of information across the organizations.

The Software and Systems Engineer's View

The key technology shifts that support this operational change are found mainly in how information is available and shared – as concepts like big data, analytics, internet of things are becoming reality and a natural part of how people engage with one another and how we look to solve problems, this will enhance our ability to work anywhere, access our work space from anywhere, and ask anyone for help.

We can say that engineers are moving from being 'in the box' in terms of ways of working, a box defined by managers and method& tools people, to being expected to act also outside of the box. Rather than following strict process, governance and metrics, and using a standardized environment they are acting in empowered teams that are focused on the outcome. We still expect the teams to be disciplined, but constantly evolving their ways of working t improve performance and use whatever they need to get the work done.

The Development Manager's View

The opportunities a manager sees in these technology shifts are typically around resource flexibility and ability to more effectively use external resources, and better tools to make predictions about cost and delivery schedules.

In terms of management style we see a shift from a command and control style with strict governance and process, to more of a collaborative leadership style where management teams are considered self-optimizing and the focus is on outcome rather than the more classic key-performance indicators, as exemplified by movement towards more agile/lean methods.

EVOLVING PRACTICE AREAS

As a consequence of changes in the engineering landscape, we see key practice areas evolving. Not necessarily revamping the way we think of these practices, but as a consequence of changes in technology and ways of working they are changing their shape. Here we are considering four areas: architecture, steering, people, and development environments.

Architecture: Model-Driven/SOA evolving to API Focused

In the past 10 years, the industry has been moving towards model-based architectures based on principles of loosely coupled systems such as service-oriented architecture (SOA). With the emergence of web and cloud-centric systems, we see a move to add the dimension of making the architectures focused on clearly defined APIs. This is reflected in a change in the way we want to do business. We want to build solutions or products that are easy to add to through well defined APIs We call this architectural style API-driven. It is interesting to note that this evolution from SOA to API-driven is not purely a technically driven change but is also one driven by the emergence of the "API economy" and an awareness that providing access to data and services represents new opportunities to monetize existing capabilities.

As part of this trend, we see movement from monolithic models and internally focused documentation to increased awareness of external "consumers" of the interfaces and focus on improving their ability to discover and use the services of our systems. In many cases, this is accompanied by the move to a more open source style of collaboration and transparency, where we design and code with the expectation that our artifacts are published and shared by a community that will include consumers outside of our organizational boundaries. We will see an increased focus on reuse of existing components open source style, vs new implementation.

Steering: Requirements Driven evolving to Outcome Focused

The classic way of steering programs and projects is through specifying and validating the implementation of requirements. There will always be requirements (regulations, standards, etc) but with the introduction of agile practices the classic requirements definition process breaks down. The focus is instead on solving the problem in more consumable chunks (i.e., "iterations") – with clear focus on achieving a business outcome vs. detailed requirements. Information access is the key to identifying and prioritizing problems. This shift is accompanied by a desire for more fluid and continuous collaboration between consumer and provider.

People: Role Based Responsibilities evolving to Team Collaboration

There will still be a need for specialized roles (technical architect, domain specialist) but the focus is on creating self-organizing cross-functional teams extended to business stakeholders and even customers and suppliers. Teams can be better equipped to self-optimize how they work by having better access to information, supported by computer-driven analytics to help interpret the vast amount of information available. We will see a higher emphasis on people and open access to information, and less on strict process or hierarchical control of information (the democratization of information).

Development and Target Environments: Integrated Standard Development Environment moving to Development & Target Environment as a Service

There is still a lot to do to better integrate tools that originally were written for specialized roles and stove-piped organizations rather than cross-functional teams. Teams expect environments to require less overhead and just work without extensive manual intervention. Development environment should transparently help teams go faster, and help remove barriers in the organization. They should not be something you need to take several days of training to understand how to use, and not something a team needs to spend man weeks to configure before they can get productive.

Considering the history of development environments, it is interesting to note that much of the more widely adopted tools are those that address communication issues appearing at organizational boundaries – think of change request management tools and requirements management tools.

There will always be a need for tools that help us organize and track engineering assets, maintain traceability, and abstract complex problems. We see a new class of tooling emerging which we call expert advisors, which help practitioners make better-informed decisions more quickly. The organization's ability to derive value from this new class of tools *depends* heavily on the quality of data available from the other lifecycle tools, and on the ability to see and extract relationships between the artifacts.

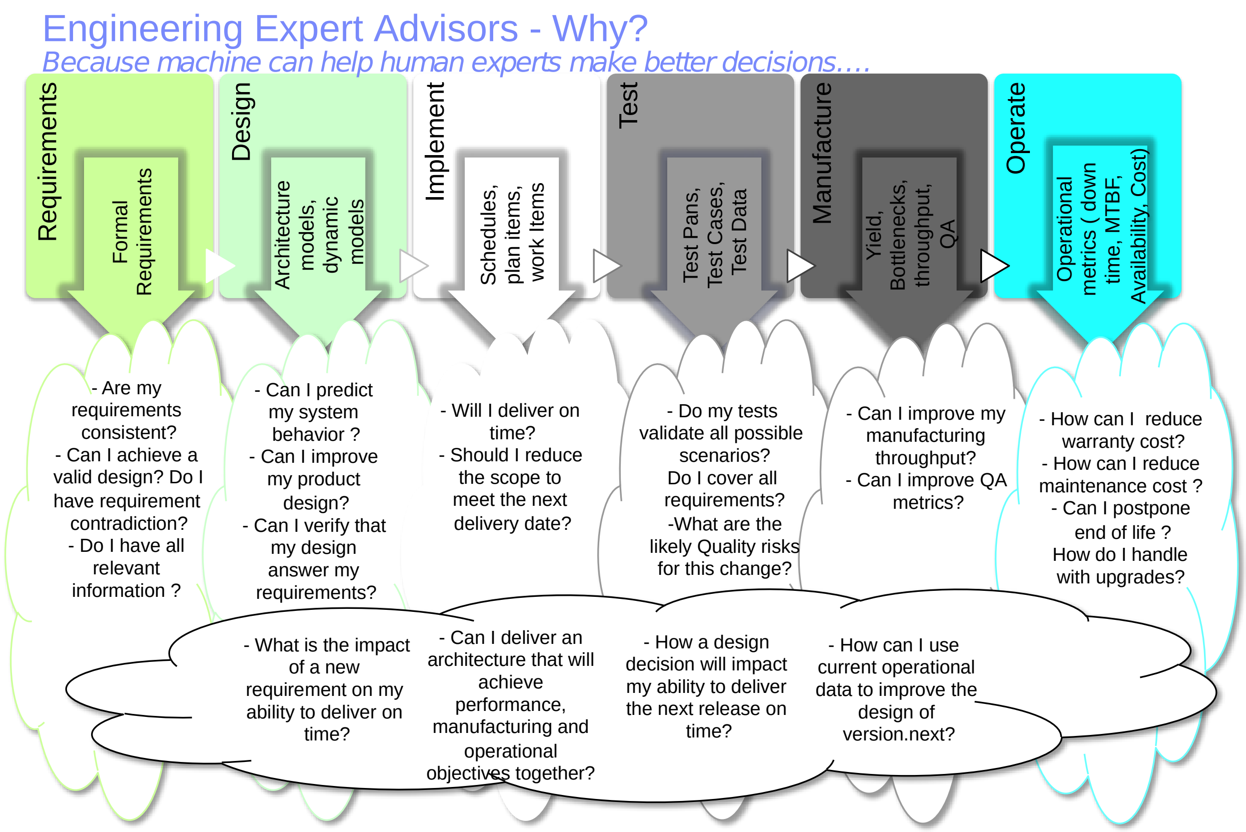

Figure 1. Expert Advisors

It is our belief that this concept of expert advisors represents the “next evolution” of Agile/Lean methods. The heart of the lean/agile methodologies is the desire for continuous learning and improvement, and the obvious frontier for improving the speed of learning is to enlist the tools to help us learn faster. Computer automation will accelerate team’s ability to continuously improve and learn, but will require that sufficient attention is paid to the instrumentation of the development process to allow analytics tools to learn from the data.

A DAY IN THE LIFE 3-5 YEARS AHEAD

Let us introduce our development team from the future. They are at work at 5GMaker Inc (a fictional mobile network service provider). 5GMaker Inc is a fictitious company name representing a manufacturer of mobile telecommunications infrastructure equipment. The development team in our scenario works for 5GMaker Inc.

Elsa is new to the development team hired just 4 weeks ago. She is 23 years old (born in 1994 – the year Netscape and WWW launched). She is brought on to a feature team for China MSP 5G customization. China MSP is a fictitious company name representing a Mobile Service Provider (MSP), deploying and operating the mobile 5G network. They are a customer of 5GMaker Inc.

Hannah is the Scrum master for the team.

Adam is the most senior architect. Unfortunately, Adam is on 2 months leave on Safari in Africa.

The development environment’s deep “trouble detection” analytics (which run continuously) have flagged an emerging problem trend in the network equipment provider’s problem reports DB and correlated with China MSP equipment logs (automatically collected with China MSP’s agreement during the early 5G rollout phase)

The development environment has correlated the problem with the deployment of the new build to a 5G base station, and identified the likely component based on cross-reference against the new features added in the most recent build and has flagged the most likely suspect components.

This information along with the links to the problem reports and the logs are included in a work item[2] created in the team collaboration function of the development environment.

Getting Started

During the daily scrum, Hannah opens the new high priority work items to review with the team and decide on how to assign. During the review they follow the data linked in the work item and see that the problem is growing in frequency, though China MSP has not escalated it yet. Hannah wants to know if they have time to investigate without disrupting the next milestone in 2 weeks time.

Looks like the only one who can work on investigating without affecting the next milestone is Elsa. Although the team is reluctant to leave this investigation to the new developer, they have no choice but to see how she handles it.

Hannah assigns the work item to Elsa. Back at her desk, Elsa opens the work item and immediately starts to investigate.

Understanding Architecture

From the work item, Elsa navigates to the feature that the Deep Analytics tool flagged as the likely origination of the problem. Elsa opens the visual Navigator and drags the feature on to it. This populates the navigator with a visual display of all the immediately related artifacts. She selects 'Build' as the target filter. The result shows the indirect traceability graph from the source (a feature) to the target (builds).

Right away, Elsa can see that this feature was first deployed in a build to US MSP[3] the previous year. She does a search for unresolved defects against Build US MSP 5G 2016 and finds none, suggesting there's something unique about the configuration of the feature in the China MSP build, OR an inadequately tested area.

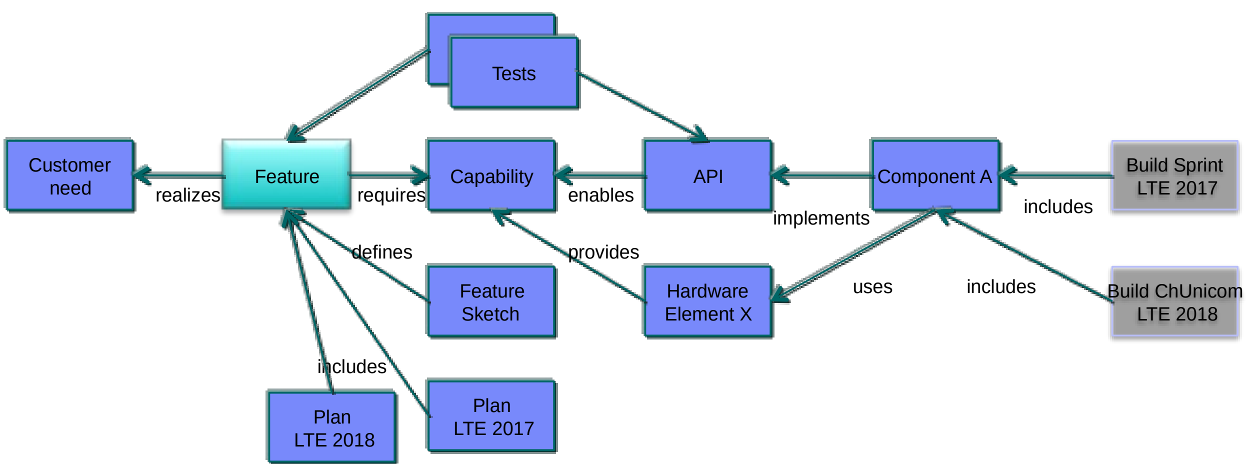

Figure 2. Understanding Architecture

Elsa can use the configuration-awareness of the visualization to see differences between these. She selects Build US MSP 5G 2016 to filter the view to see the specific versions of these artifacts as used in that build.

[2] a "work item" is a task that needs to be done by a team as part of a larger project. The work item is the electronic record of that task, the people who worked on it, the states or stages of process that it must go through in order to conform (and prove conformance) to the development process for the organization, and links to artifacts required or modified to accomplish the task.

[3] Another fictitious company name representing a US Mobile Service Provider.

She saves a snapshot of this view.

She then generates a view for China MSP 2017 build and does a comparison to see what is different. It appears that newer version of a Hardware chip was used and Elsa identifies this as the possible root cause.

Elsa checks the Quality 'Expert Advisor' for 'ChMSP2018' and sees that the advisor warns that defects are 78% likely in this feature due to this change. Perhaps the Quality Advisor feature was not available when they did this build.

Show versions: filter configuration "US MSP 5G 2016"

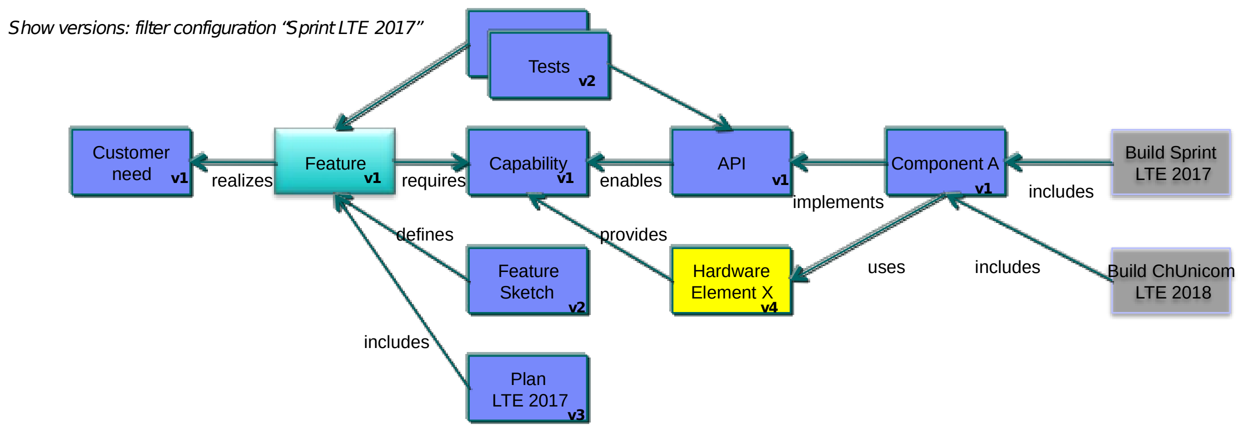

Figure 3. Analyzing Architecture

Social Discovery

She wants to review the problem logs with someone who understands how the feature tests were written. If they can reproduce the problem in their lab they should be able to isolate it quickly. She goes back to the navigation window and selects the test cases then selects a 'related experts' button that examines the relationships to the users who most frequently updated this artifact or immediately related artifacts such as work items.

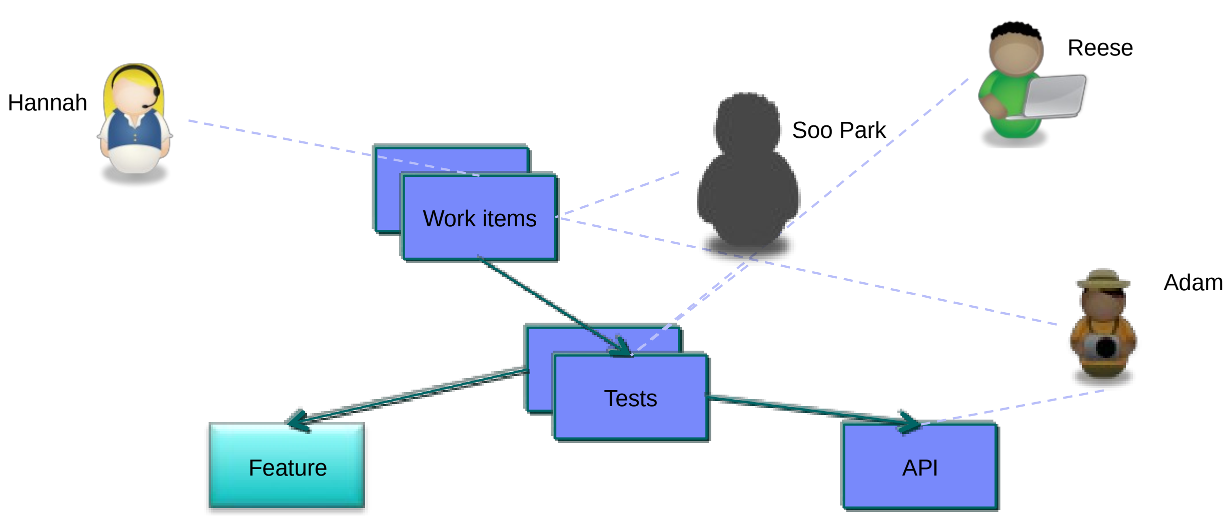

Figure 4. Social Discovery

This view combines social awareness and topic knowledge. The size of the team member icon shows how frequently and recently they worked on the topic. The coloration shows their availability. Soo Park was the most knowledgeable about the tests, but is no longer with the company. Reese is the next most knowledgeable. She talks to Reese about the tests. Reese guides her on which tests will be necessary and how to set up the test run.

Elsa does a quick search on the test lab environment looking for any nodes equipped with v4 of the Element.X chip. She finds one node in one of the test racks.

She schedules a test run of the existing tests for this feature through the integrated test generation equipment on Test Rack 1/Node 3 using the ChMSP2018 build.

The tests pass but analysis of the logs shows the same anomaly. The logs seem to show that a timeout is occurring in one of the message responses expected from Node 3.

She writes a new test case variant but changes one of the timing parameters to change expected response timing, and then schedules another test run. This time the test fails exactly as expected. Something about the hardware change has subtly changed the timing.

She looks up the part in the part tree and initiates a discussion thread on the chip. Since the chip comes from a supplier, the question and any follow-ons are automatically emailed to the supplier. The supplier does have limited access just to this part's data, its requirements and associated tests.

Mobile Access

This supplier is based in the US. Elsa is hopeful she gets some response prior to the next day's daily scrum.

Later in the day in the US, the supplier confirms in the thread that the on-chip buffer design was changed and it is possible that could introduce delay in software receiving the response signal.

Elsa gets an immediate notification on her iPhone client that the thread has new responses. Her iPhone securely connects to her corp. system once she verifies with her thumbprint.

She looks up the unit tests associated with the Hardware Element to see if she can run a test that would isolate or prove the timing problem

Elsa finds the hardware element using a simple keyword search. The element contains links to the unit tests. Sarah loads the unit tests to her workspace. The workspace wizard identifies likely dependent components to load, Sarah accepts recommended dependencies and loads them to workspace.

She explores the unit tests but does not find one that tests for a longer latency message. She uses one of the existing tests to write up a new test that will test for the latency issue using the mobile version of the IDE.

On her mobile she finds the test lab equipment that has the right Hardware Element configured. She uses the test scheduler to set up the test run for later that night.

Work Fix into Sprint Plan

The next morning Elsa's test run results appear at the top of her main page. The result does show that there is a timing window that can cause some messages to exceed the specified latency. Elsa links the chip design discussion thread, and her new unit test and the test result back in the work item and closes the work item with a recommendation to try updating the component to do an additional hardware check on the buffer when the timeout occurs per the chip supplier's recommendation.

During the team's daily scrum the next day, Elsa reports her findings.

Hannah immediately tries to check the impact to project plan if they squeeze the fix in the next sprint with Elsa as the lead developer.

The Planning Expert Advisor provides an indication that this plan has low probability of success. The reason is the changes to Component A have historically taken longer than estimated by a large margin. AND, historically changes made by someone who has never worked on a Component before result in higher % of defects and more time on test and defect fixing.

Hannah tries a ‘what if’ scenario assuming she can get help from another developer on a different feature team who worked on Component in the past. Elsa tells her that Reese has experience in this area. With that person assigned to help Hannah with the change, the plan has a reasonably high probability of success. She also creates a plan dependency on ‘Reese’.

Hannah assigns the work item for the next iteration to Elsa, and sends a high priority request for some of Reese’s time to his manager.

CONCLUSIONS

Key technology shifts in cloud, analytics, mobile and social are all impacting the way engineering organizations operate. They have a fundamental impact on our ability to comprehend and analyze engineering data. An observation is that engineering and software organizations seem to be one of the last bastions not adopting and taking advantage of analytics technology – there are significant opportunities for productivity and quality improvements from the use of analytics. Also, the democratization of information and the expectation of young engineers is that you can get a first approximation of answers to your problems from 'the environment' (applying analytics and social techniques) and be less dependent on interaction with your local 'guru', and to be able to do it 'anywhere' not only in the confines of the your cubicle.

From an engineers perspective, the key words in this transformation are empowerment and flexibility. Managers will look for improved productivity, more resource flexibility and better predictions.

REFERENCES

Bradicich, Tom. Orci, Stephanie. (2013), NI.com, “The Moore’s Law of Big Data”. http://www.ni.com/newsletter/51649/en/

Denning, Stephen. (2010), “*The Leader's Guide to Radical Management: Reinventing the Workplace for the 21st Century*”. Jossey-Bass, a Wiley Imprint, San Fransisco, CA.

Denning, Stephen. (2012), “*The Best-Kept Management Secret On The Planet: Agile*”. Forbes, http://www.forbes.com/sites/stevedenning/2012/04/09/the-best-kept-management-secret-on-the-planet-agile/.

IBM Research. (2014) IBM.com, “ANDES: Predicting the Likelihood of On-Time Delivery”. http://researcher.ibm.com/researcher/view_project.php?id=4402

IBM. (2014). IBM.com: Explore IBM JazzHub projects. https://hub.jazz.net/explore

Ries, Eric. (2011), “*The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*”. Crown Business, a division of Crown Publishing Group Inc, New York, NY.

Speicher, Steve. “Overview of Linked Data and OSLC”, Open Services for Lifecycle Collaboration wiki. http://open-services.net/resources/videos/overview-of-linked-data-and-oslc/

Thomas, Owen. (2014), “How APIs Connect The World”. Readwrite.com. http://readwrite.com/2014/03/14/api-explainer-intel#awesm=~oA86YofwdiojpP

Womack, James P., Jones, Daniel T., Roos, Daniel (1990), “The Machine That Changed the World: The Story of Lean Production-- Toyota's Secret Weapon in the Global Car Wars that is Revolutionizing World Industry”. Rawson Associates Scribner, Simon & Schuster Inc. New York, NY.