A Lean Approach to Scheduling Systems Engineering Resources

Dr. Richard Turner, Stevens Institute

Abstract. Integrated Master Schedules serve a critical purpose in coordinating system development. However, in large operational systems where evolution is both rapid and externally driven, they can fail to provide the flexibility and visibility required for strategic decision making. Research into applying lean concepts to scheduling may hold the answer.

Understanding the status of evolutionary capability development in large operational systems is often difficult. Schedules are rarely stable due to:

- Size and complexity of capabilities
- Operationally imposed unexpected changes in priorities
- Deep supplier chains and contract structures
- Variety and availability of special engineering resources
- Generally complex nature of the operations.

This instability can cause undue stress on traditional scheduling models and tools. The effort required to maintain large networks and plans—exemplified by Integrated Master Schedules and Plans, complex work breakdown structures and earned value management systems—often outweighs their usefulness for communicating progress and managing resources. It is difficult to understand at any particular point in time:

- How much work the organization, program or project resources have the current capacity to perform within a specified time frame
- What resources are overcommitted or underutilized
- What work is actively proceeding
- How much work is actively proceeding
- What work is blocked and thus inactive
- What is the continuing impact of unpredicted changes in priority, urgent maintenance or critical development responses, changes in requirements or scope, or other emergent issues
- What is the actual progress toward various project outcomes
- How often and when is value finally delivered to the user

Lean approaches strive to maximize flow through a process—often by using on-demand (Kanban) scheduling techniques. Software development organizations have found that iterative and on-demand approaches are more flexible and provide better results than traditional push scheduling methods. The Systems Engineering Research Center (SERC), a University-affiliated research center of more than 20 universities and research organizations led by Stevens Institute, is investigating on-demand scheduling techniques in SE to determine if they can provide:

- Better status visibility managing multiple concurrent development projects
- More effective integration and use of scarce SE resources
- Increased project and enterprise value delivered earlier
- More flexibility while retaining predictability
- Less blocking of product team tasks waiting for SE response
- Lower governance overhead.

To investigate these aspects, SERC researchers have identified large, evolving, software-driven systems as the target environment for their initial work. These systems make up a significant portion of defense acquisitions, and include complex real-time actions that often occur across systems of systems.

After studying the needs of several government and commercial environments, the SERC team has combined several agile and lean ideas and is experimenting with their application in SE. We have defined a general Kanban-based Scheduling System (KSS) that we believe captures the essence of lean flow management visibility and flexibility. We have also developed a concept for SE as a service to support ongoing collaboration between SE and SW tasks. Together, we postulate that these approaches will improve the ability to reallocate resources as needed to meet emergent needs, continue to support ongoing development and maintenance activities, all without overloading resources. We are now describing and simulating the implementation of a network of KSSs in a complex, multi-site health care information system.

The KSS Concept

A KSS is a means of visually controlling workflow. It consists of a set of activities, where each activity has its own ready queue, a set of resources to add value to work units that flow through it, and a done queue.

Visual representation provides immediate understanding of the state of flow through the set of activities. This transparency makes resource issues and process anomalies (both common and special cause) easily visible, enabling the team to recognize and react immediately to resolve issues locally. Because the team and management interact with the visualization and collectively solve problems, this aspect is important in achieving continuous improvement (Kaizen). Control of the KSS is generally maintained through Work in Progress (WIP) limits, small batch size, and Classes-of-Service (CoS) definitions that prioritize work with respect to risk.

WIP is partially completed work, equivalent to the manufacturing concept of parts inventory waiting to be processed by a production step. WIP in knowledge work can be roughly associated to the number of work items started and not delivered. WIP Limits specifically cap the amount of work assigned to a set of resources. This lowers the context-switching overhead that impacts individuals or teams attempting to handle many simultaneous work items. WIP Limits accelerate useful value by completing work in progress before starting new work and also provide for reasonable and sustainable resource work loads.

CoS provides a variety of handling options for different types of work items and influence the next task selection within KSSs. They allow the WIP limits to be distributed in such a way that certain types of work will always take priority, will have more consistent access to resources, or will only be selected under certain circumstances.

The fundamental KSS building block is shown in Figure 1. In general, the upstream customer for the service provided is responsible for selecting the work that enters the KSS. This is usually done collaboratively with the KSS to make sure that significant dependencies, date-certain events, and other special concerns are understood. As resources become available, the highest value work item is selected, resources are assigned, the work is executed until it is complete, and then added to the completed work queue. The value of a work item depends on a number of factors, including priority of the project associate with the work, cost of delaying the work, criticality of the work, and the work's impact across the larger system or system of systems. A scheduling cadence provides regular meetings of the KSS team to assess flow and determine if resources should be moved between activities, WIP limits adjusted, or other actions taken. Often, this is a daily activity, but the actual planning horizon selected and the nature of the work items should be used to establish the most cost-effective cadence.

SE as a Service

Defining SE as a service and using on-demand scheduling is designed to better allocate scarce SE resource and integrate the SE flow with the SW development project flow. If SE is seen as an overarching and somewhat separate activity, there is little ability to interact with the needs of the various development teams, and no means of identifying priority when asked for support. SE and developers both have unique insights into the rationale and reality of the systems or products they are evolving but often do not realize how their decisions impact their counterparts or the system as a whole. Viewing SE as a service performed for management or product/system developers generates communication opportunities that enable negotiation and collaboration in determining the priority, scheduling, and quality level of technical activities.

In general, SE is involved in three kinds of activities in rapid response environments: lifecycle, continuous, and requested. Lifecycle activities are critical in greenfield projects, but are important in all systems and system of systems evolution. They include front-end work like creating operational concepts, defining architectures, and capability and requirement decomposition and allocation, as well as final verification, validation, integration, and deployment activities. Continuous activities are ongoing, system–level activities (e.g. architecture analysis, performance analysis, configuration and risk management, and incremental verification, validation and integration). These require regular resources for analysis and the maintenance and evolution of long-term, persistent artifacts that support multiple projects. Requested activities are generally specific to either individual projects or capability engineering (e.g. issue triage, trade studies, impact assessments, needs analyses, cost analyses, interface support, and specialty engineering support), and draw on the persistent SE artifacts and knowledge.

By viewing persistent artifacts (architectures, requirements, interfaces specifications) as key components of services provided to various projects, SE can be opportunistic and apply its cross-project view and its understanding of the broader environment to better support specific projects individually or in groups. It can also broker information between individual projects where there may be contractual or access barriers. When a system-wide issue or external change occurs, SE can negotiate or unilaterally add or modify work items within affected projects to ensure that the broader issue is handled in an effective and compatible way. The quality of a service may be pre-specified, specified as a parameter of the service request, or negotiated as a function of typical value sought and time available to provide the service. SE services may be thought of as a single activity, although many activities are complex enough to have their own set of value-adding activities and specialized resources.

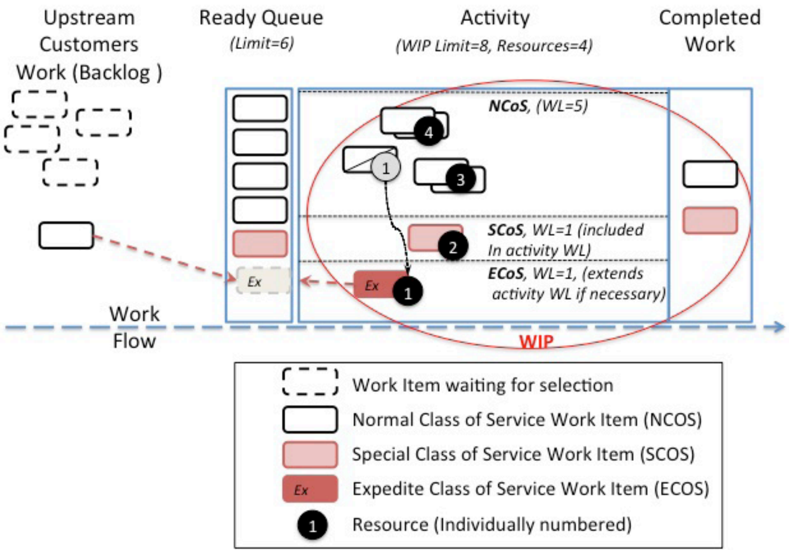

Upstream Customers Work (Backlog)
Ready Queue
(Limit=6)
Activity
(WIP Limit=8, Resources=4)
Completed Work
NCoS, (WL=5)
4
1
3
SCoS, WL=1 (included In activity WL)
2
ECoS, WL=1, (extends activity WL if necessary)
Ex
1
Work Flow
WIP
Work Item waiting for selection
Normal Class of Service Work Item (NCOS)
Special Class of Service Work Item (SCOS)
Ex
Expedite Class of Service Work Item (ECOS)
1
Resource (Individually numbered)

Figure 1. KSS Building Block

To support timeliness, SE performs its services in parallel to those activities in the requesting project. SE can use the KSS network constructs to compare the values of individual project work across the entire system and select the most critical work (often the work presenting the highest cost of delay) to accomplish next. This increases the effectiveness of the limited SE resources across the enterprise.

Implementing a KSS Network

The initial implementation uses a network of integrated KSSs that are intended to:

- Shorten the time required to deliver value to internal and external consumers
- Make work in progress and status visible at all levels through Dashboards and KSS flow boards
- Monitor organizational capacities at all levels
- Support analysis and decision making at every level of management
- Limit WIP to improve value flow (identify resource issues, cause of blocked work)
- Coordinate multiple levels of SE activity; enable cross-organizational teams and swarming of resources
- Establish a basis for continuous improvement in a rapidly changing environment

The KSS Network shows the relationships between the SW development tasks and the SE tasks. It also clearly captures the relationships between the SW and SE tasks and the capabilities. Understanding the information needs for decision making, including scheduling and flow monitoring/control, at each level of SE activity or utilization, is a key to a successful KSS design. Figure 2 shows the current conceptual design of the hospital system KSS Network.

The KSS Network acts as a distributed database of changing work status and value. This provides a basis for informed decision making at every level and encourages pushing technical decisions to the lowest level appropriate. A critical characteristic is the transparency provided by the near-universal status availability and the specificity of the policies that underlay the scheduling decisions. These policies are most often defined using CoS, WIP limits, and value definitions, and are exercised by informed, collaborative decision makers.

The CoS that have been identified for the heath care system KSS Network are presented in Table 1. The definition of initial WIP Limits, collaboration mechanisms for specific types of work items, and value determination algorithms are still underway.

Conclusions

This research has included a great deal of wandering through exciting possibilities and running into stark realities. The concepts are reasonable, but the work in applying them to the difficult environments we have chosen is just beginning. The next step is to simulate the example health care LSS Network and experiment with the mechanisms we have defined to implement its activities. This is not a simple task.

To support both the concept evolution and the simulation development, we are also looking for interested organizations or projects to pilot single or multiple level LSS concepts on their own work flow. The data gathered and experience provided by such pilots would be extremely helpful to the research, but may also support the pilot organizations in beginning the culture change initiatives that inevitably accompany transition.

Acknowledgements:

This material is based upon work supported, in whole or in part, by the DoD through SERC under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. The SERC RT-35 Research Team includes Richard Turner, PI, Stevens Institute; Ray Madachy, Co-PI, Naval Postgraduate School; and Barry Boehm, Jo Ann Lane, and Dan Ingold, University of Southern California.

Additionally, a volunteer Industry Working Group has been essential to formulating the approaches with an eye to the needs of the work environment. Its members include David Anderson (David J. Anderson and Associates), Jabe Bloom (The Library Corporation), Hillel Glazer (Entinex), Curtis Hibbs (Boeing), Suzette Johnson (Northrop Grumman), Larry Maccherone (Rally Development), Don Reinertsen (Reinertsen & Associates), David Rico (Boeing), Garry Roedler (Lockheed Martin), Karl Scotland (Rally Software, UK), Alan Shalloway (NetObjectives), Neil Shirk (Lockheed Martin), Neil Siegel (Northrop Grumman), and James Sutton (Jubata Group).◆

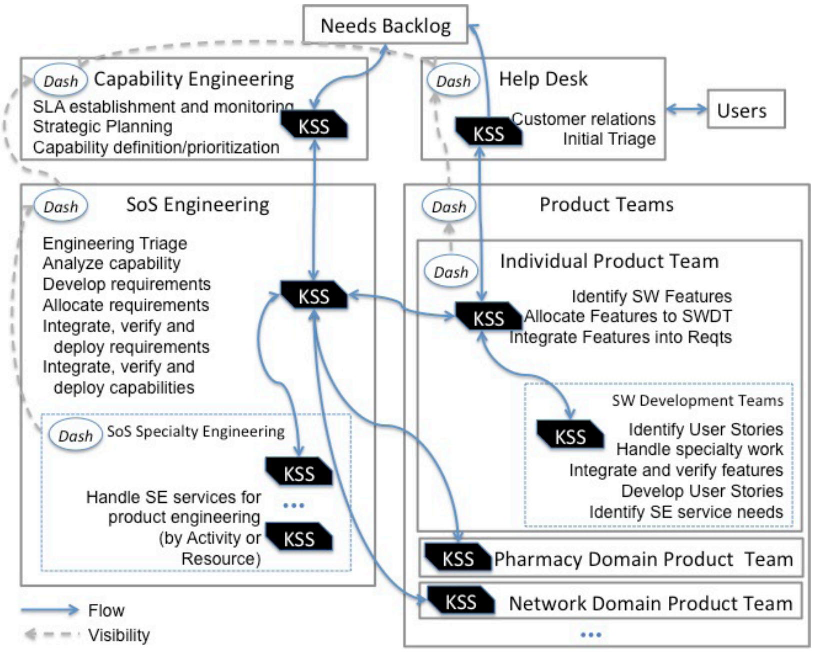
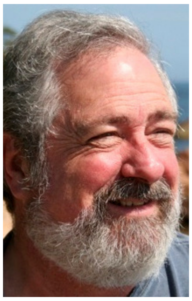

Needs Backlog
Dash
Capability Engineering
SLA establishment and monitoring
Strategic Planning
Capability definition/prioritization
KSS
Dash
Help Desk
Customer relations
Initial Triage
KSS
Users
Dash
SoS Engineering
Engineering Triage
Analyze capability
Develop requirements
Allocate requirements
Integrate, verify and deploy requirements
Integrate, verify and deploy capabilities
KSS
Dash
SoS Specialty Engineering
KSS
Handle SE services for product engineering (by Activity or Resource)
...
KSS
Dash
Product Teams
Dash
Individual Product Team
Identify SW Features
Allocate Features to SWDT
Integrate Features into Reqts
KSS
SW Development Teams
KSS
Identify User Stories
Handle specialty work
Integrate and verify features
Develop User Stories
Identify SE service needs
KSS
Pharmacy Domain Product Team
KSS
Network Domain Product Team
...
Flow
Visibility

Figure 2. Healthcare LSS Network

Standard/enterprise-wide Classes of Service	
Critical Expedite	A Critical Expedite work item represents something that fixes an existing or imminent issue within the system. Safety, security, or other emergency work items are assigned this CoS. It is disruptive and requires all appropriately skilled resources to suspend their current activities and work on the Critical work item. It also suspends any WIP limits in activities associated with its work items for the duration that the critical work is in the activity. Once a work item is assigned this CoS, the CoS applies to all derived work items in all KSSs, regardless of local priorities.
Important	The Important CoS is assigned to very high priority work items where the speed of completion is such that this work should take priority over all other work in the ready queue. It is not disruptive, because all WIP is allowed to finish before the important work begins. It does not impact WIP limits, but has a guaranteed WIP limit in some KSSs.
Date Certain	Date certain (or schedule as independent variable) class of service reflects work items that must be completed by a specific date or there will be significant consequences. Regulatory implementation deadlines, COTS upgrade preparation, or integration/deployment dependencies are candidates for this class of service. It operates essentially like an Important CoS, but as the date becomes closer, it may elevate to Critical Expedite based on workload.
Standard	This is the normal CoS for the development organizations work. A high percentage of work should be assigned at this level for the KSS Network to provide the desired outcomes.
Background	Background work (sometimes referred as intrinsic or invisible) is work that must go on but is usually not time critical. It includes things like architectural enhancements, low-level technical debt, research and environmental scanning, or time-certain events not due in the near future. It is usually prioritized by its length of time in the queue (FIFO). Some KSSs may have a limit for the time background work can remain in the queue. When reached, this limit automatically triggers placement of the work item in a higher CoS.
Special Limited Classes of Service	
Collaborative	This special CoS is designed for activities that span organizations and organizational levels such as cross-discipline issue analysis. The actual mechanisms for implementing this CoS are not complete, but it will be designed so that shared resources can be tracked across multiple LSSs without changing the basic flow management and scheduling activities.
Product Support	This CoS is limited to certain SE LSSs that directly support Product Team requests. It is designed to limit the impact of other classes of service on work items resulting from product teams requests. There is a guaranteed WIP Limit allocation for this work, meaning there are always resources allocated to this class of work, preventing the complete blockage of flow. It also raises the priority or value of work over time similarly to the Background CoS.

Table 1. Initial CoS for Health Care Systems KSS Network

ABOUT THE AUTHOR



Dr. Richard Turner has more than thirty years of experience in systems, software and acquisition engineering. Currently a Distinguished Service Professor at the Stevens Institute of Technology in Hoboken, New Jersey, he is co-author of three books: Balancing Agility and Discipline: A Guide for the Perplexed, co-written with Barry Boehm, CMMI Distilled, and CMMI Survival Guide: Just Enough Process Improvement. Dr. Turner is a Fellow of the Lean Systems Society.

E-mail: rturner@stevens.edu

RECOMMENDED READING

1. Boehm, Barry and Turner, Richard (2004). Balancing Agility and Discipline: A Guide for the Perplexed. Boston, MA: Addison Wesley.
2. Larman C. and Vodde, B. (2009). Scaling Lean & Agile Development. Boston, MA: Addison Wesley.
3. Poppendiek, Mary. (2007). Implementing Lean Software Development: Boston, MA: Addison Wesley.
4. Turner, Richard and Wade, J. (2011). Lean Systems Engineering within System Design Activities, Proceedings of the 3rd Lean System and Software Conference, May 2-6, 2011, Los Angeles, CA.
5. NDIA-National Defense Industrial Association (2010). Top Systems Engineering Issues In US Defense Industry. Systems Engineering Division Task Group Report, <http://www.ndia.org/Divisions/Divisions/SystemsEngineering/Documents/Studies/Top%20SE%20Issues%202010%20Report%20v11%20FINAL.pdf>, September, 2010.
6. Turner, Richard, Shull F., et al (2009a) "Evaluation of Systems Engineering Methods, Processes and Tools on Department of Defense and Intelligence Community Programs: Phase 1 Final Technical Report," Systems Engineering Research Center, SERC-2009-TR002, September 2009.
7. Turner, Richard, Shull F., et al (2009b) "Evaluation of Systems Engineering Methods, Processes and Tools on Department of Defense and Intelligence Community Programs: Phase 2 Final Technical Report," Systems Engineering Research Center, SERC-2010-TR004, December 2009.
8. Anderson, David. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Sequim, WA: Blue Hole Press
9. Burrows, Mike. (2010). Kanban in a Nutshell. Blog post. <http://positiveincline.com/index.php/2010/03/kanban-in-a-nutshell/>, March, 2010.
10. Reinertsen, Donald G. (2010). The Principles of Product Development Flow. Redondo Beach, CA: Celeritas Publishing.
11. Boehm, B. et al. "Applying the Incremental Commitment Model to Brownfield Systems Development," Proceedings, CSER 2009, April 2009.
12. Boehm, B., and Lane, J., "Using the ICSM to Integrate System Acquisition, SE, and Software Engineering," CrossTalk, October 2007, pp. 4-9.
13. Turner, R., Madachy R., Ingold D., and Lane J., "Modeling Kanban Processes in Systems Engineering," submitted to International Conference on Software and System Process 2012, 2012.
14. Turner, R., J.A. Lane, D. Ingold, R. Madachy, and D Anderson. "An event-driven, value-based, pull systems engineering scheduling approach." Systems Conference (SysCon), 2012 IEEE International,. IEEE, 2012. 1-7.
15. Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering (2008). Systems Engineering Guide for Systems of Systems, Version 1.0. Washington, DC: ODUSD(A&T)SSE, 2008.

CROSSTALK
Nov / Dec 2012 The Journal of Defense Software Engineering Vol. 25 No. 6
CROSSTALK
Jan / Feb 2013 The Journal of Defense Software Engineering Vol. 26 No. 1
SOF
PRO
MAN
LESSO
CROSSTALK
March / April 2013 The Journal of Defense Software Engineering Vol. 26 No. 2
Supply Chain Risk Management

CALL FOR ARTICLES

If your experience or research has produced information that could be useful to others, CrossTalk can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

Real-Time Information Assurance
Nov/Dec 2013 Issue
Submission Deadline: June 10, 2013

Please follow the Author Guidelines for CrossTalk, available on the Internet at <www.crosstalkonline.org/submission-guidelines>. We accept article submissions on software-related topics at any time, along with Letters to the Editor and BackTalk. To see a list of themes for upcoming issues or to learn more about the types of articles we're looking for visit <www.crosstalkonline.org/theme-calendar>.