Transdisciplinary Engineering: Crossing Boundaries
M. Borsato et al. (Eds.)

doi:10.3233/978-1-61499-703-0-299

Scope Patterns for Projects Modeled as Sociotechnical Systems

Bryan R. MOSER[a, b,1]
[a] *Massachusetts Institute of Technology*
[b] *University of Tokyo*

Abstract. This paper examines the treatment of scope as project attribute, arguing that an improved representation will allow increased fidelity of project plan trade space enumeration and preferred plan selection. Cost, schedule, and scope are emergent characteristics of a project's integrated architecture, activities, and resources uniquely for a project at hand. System engineering as commonly practiced places strong, early emphasis on product architecture and requirements, including enumeration of system options prior to interplay with aspects of project implementation. As such, system options are often framed and pruned prior to effective examination of project feasibility. Characteristics of scope are presented suitable for model-based design of projects. Scope is defined as the tangible outcomes of project tasks. Scope items should be useful in the evolution of project knowledge and interplay with requirements and resources. Target-neutral, resource-nominal, and exception-capable patterns of scope are described.

Keywords. Model-based systems engineering, scope, triple constraint, scope patterns, sociotechnical systems, project design, MBPD

Introduction

In engineering much attention has been given to consideration of stakeholder needs and their relationship to product requirements. In best practice, these requirements are written as solution neutral, so that a trade space of product solution alternatives may be enumerated prior to generation and evaluation of options, unbiased by prior assumptions [1].

Typically, a similar neutrality is not maintained for consideration of *project* alternatives. On the contrary, upon selection of architectural options it is common to determine the utility of product system alternatives prior to consideration of resource and other implementation issues. *How* the system is to be implemented is considered a separate analysis from *what* the system should be. In those cases where feasibility is considered, assumptions of likely implementation reply upon historically based process cost, schedule and quality or a notional standard work. Deeps assumptions from a century of scientific management reinforce this reliance upon implementation as operational repetition with low variation [2].

We have seen therefore the enumeration of product system trade spaces without interplay with project options, thus biasing and pruning alternatives prior to effective examination of feasibility. An optimal system architecture and technical approach is

[1] Corresponding Author, E-Mail: bry@mit.edu

selected, yet later to be discovered infeasible within cost, schedule and quality implementation constraints. Given complex organizations and the burden of document-based workflow this separation has been understandable. With the extension of model-based systems engineering (MBSE) to the design of projects, an opportunity is presented to concurrently retain options both desirable and feasible. *The product system and project system should be designed together*.

In the development of model-based project design (MBPD), this paper explores a definition of scope useful in project models. When viewing a project as a sociotechnical system, cost, schedule, and quality are emergent characteristics of the scope, resources, and architecture, often unique to the project at hand [3]. In the spirit of concurrent engineering, both product and project systems alternatives should be retained until sufficient knowledge is gained through investigation and decisions.

This paper focuses on characteristics of project scope, an attribute of a project model with useful evolution and interplay with other project attributes. Examples of target-neutral, resource-nominal, and exception realistic patterns of scope are shown. This approach improves comparison of scope across projects and enables more realistic forecasting at the start and as a project changes.

This paper begins with a review of related research, introduces a definition of project scope suitable for model-based project management, describes characteristics of scope units and patterns, and discusses several examples. The paper concludes with known limitations and next steps in this research.

1. Conventional Representations of Scope

A review of related literature shows a wide range of uses of the term “scope” in project management. Most commonly scope refers to the boundaries and content of a project: what work is included and what work is not included. However, *scope* varyingly refers to project targets, requirements, processes, deliverables, resources, and even milestones and schedule dates. Based on this wide range of interpretations, one could conclude that the scope IS the project, rather than one aspect or dimension of the project.

1.1. Project Management Standards and Practices

Let’s begin with the standard definition from the PMI’s Project Management Body of Knowledge (PMBOK): “Project scope. The work that needs to be accomplished to deliver a product, service, or result with the specified features and functions” [4]. Given this definition, several interpretations are reasonable in practice. Does scope refer to work as process, the outcome of tasks, or even the architecture of the project? Is scope the project work itself or the activities related to managing the project? What attributes of a project are not an aspect of scope?

Even within the PMBOK other references to scope and scope management are inconsistent. Scope management is defined to include “the project and product requirements, criteria, assumptions, constraints, and other influences related to a project, and how each will be managed.” Here one can see an expansive definition that would seem to include everything both internal to and influencing a project. This language in PMBOK is readily copied throughout the internet and practice; a simple internet search shows language about scope management repeated over and over, yet applied diversely. Many follow-this pattern consistent with PMBOK: claiming the importance of scope definition and management, providing an expansive but in the end inconclusive definition [5].

Experience in the field with complex engineering projects bears out this confusion. Very often dysfunctional dialogue amongst key teams during design and implementation is rooted in misplaced uncertainty and misalignment given different understanding of scope in the project. If one is to include scope as an attribute in a project model, a more specific definition of scope is needed.

1.2. Scope as Part of the Triple Constraint

In practice, and across various standards, scope is referred to as one of three dimensions of a project to be managed: scope, cost and schedule. As a project is planned and managed over time, these three dimensions are coupled: at some point the increase in scope will impact cost or schedule, and so on. Informally these three dimensions are also referred to as the *iron triangle*, in that once a project has been optimized in may not be feasible to improve all three dimensions at the same time. More recent versions of PMBOK have expanded the original triple constraint with three more elements: Scope, Quality, Schedule, Budget, Resources, and Risk.

Lee-Kelley, pointing to the well-known Handbook of Project Management by Turner [6], referred to scope as one of five dimensions of a project, along with organization, cost, time, and quality. Consistent with the broader usage the author wrote "the scope of the project demarcates its work boundary and is managed through the product and work breakdown, which are derived from a 'hierarchy' of objectives from 'vision, mission, facility, team and individual objectives" [7]. Lee-Kelley proceeded though to refer to scope without formal definition, implying that scope is a measure of the project duration, team size, and skills required. As in other references, "scope" seems to act a stand in for many aspects of a project.

Across these approaches, whether scope is one of three, five, or six dimensions of the project, an underlying premise is that scope is not the same as these other dimensions, especially cost and schedule.

1.3. Task Estimation based on Interplay of Scope and Resources

Others have considered the nature of scope in order to estimate task size. Brooks famously argued that use of *man-month* as a unit for measurement of the size of a job is "dangerous". He wrote that "Man and months are interchangeable commodities only when a task can be partitioned among many workers with no communication among them"[8]. Brooks added that traditional estimates also ignore the likely yet unexpected difficulties in execution. Importantly, when considering scope Brooks weighs the nature of particular kinds of activities specific to software development, rather than generic tasks of fixed duration, and suggests estimation based on these characteristics while avoiding linear extrapolation of small, task-local estimates to larger projects.

Similar to Brook's mythical man-month, Lanigan exposed the fallacy of estimation without considering both the nature of the task and the characteristic of resources [9]. Lanigan proposed characterization of a task's scope with nominal effort description of the task including a minimal team size, beyond which linear extrapolation would not make sense.

Rodriguez empirically explored the trade-off of team size and productivity for software projects [10]. Based on a large data set of benchmarked I.T. projects, the research objective was to uncover productivity trends by scoring the scope and effort of these projects. They attempted to derive concrete units of scope from overall project sizing and count of designed units or function points, and productivity as function points per staff hour. Results were consistent with a premise that teams of nine or more have diminishing returns in productivity.

Holtta-Alto and Magee explored human resource utilization in product development projects across several sectors. They found that for large projects resource factors are often masked since estimation is conducted at a higher level. They also pointed out that for many estimation approaches rarely are interactions taken into account [11]. Both Holtta-Alto and Rodriguez point out the insufficiency of typical methods which rely on data from past projects which resource issues, interactions, and strife with missing values and errors.

1.4. Complexity, Readiness, and Standard Units

Clark in 1989 referred to project scope in product development as an aspect of strategy: the amount of new content developed in-house. A project with off the shelf mature parts is described as having less scope than a project with parts that require new and uncertain development efforts. Parts that are outsourced to suppliers might require the supplier to engage in more scope, yet that scope is outside the boundary of the project as defined in Clark's representation [12].

In addition to the degree of new content, some industries manage projects which require larger commitments prior to starting and significant risks of changes. In these industries an increase in planning to determine scope offsets downstream consequences of getting scope wrong.

The Product Definition Readiness Index (PDRI) was developed in the 1990s by the Construction Industry Institute (CII) to improved scope definition in capital projects. While PDRI is weighted scorecard to audit the readiness of a capital program to proceed to a major gateway, PDRI covers more than scope. The scorecard includes a broad range of items for readiness including strategy, requirements, site information, procurement processes, equipment, resources, control, risks, and so on. Therefore, PDRI -- even though described as a scope management tool -- refers to the readiness of the entire project for implementation, consistent with a broad definition of scope as project. Other than the long list of items in PDRI, a clear definition of "project scope", however, is not given [13] [14].

A recent approach to scope for the construction industry was proposed by Song [15], named the quantitative engineering project scope definition (QEPSD). Their paper describes scope as "subdividing the overall project deliverables into smaller and more manageable components, resulting in better project planning and control." They report that in construction projects previously scope could have been estimated by the number of design documents for each functional area. However, due to the shift to digital (CAD models) for the as-built environment these measures are no longer relevant. Instead Song recommends units more closely tied to physical design deliverables, referred to as "design items" in functional terms, adjusted for complexity. Within a functional area, for example structural steel, the discipline should define standard units as an abstract measure against which specific jobs can be compared.

These authors repeat a common caution from those relying on past data in studies: many corporations have tracked progress as aggregate measures of staff hours against account codes as a stand in for scope. Similar to arguments against EVMS, the existing data available was leveraged as a stand in for broader project characteristics, requiring heavy assumptions leading to inaccurate understanding of these non-cost related characteristics. These past cost or effort data alone should not be confused with scope.

2. Scope

By review of related literature, we can see that the term *scope* has been used broadly, with intermingling of ideas relevant to project targets, requirements, resources, architecture, and cost/schedule. We ask, at what point is the characteristic an aspect of scope rather than an aspect of value, of resources, or of architecture? How can we describe the scope so that resources can be matched, yet not presume what the specific resource will be? If we change non-scope characteristics in a project, yet leave scope the same, will a change in total project outcomes be observable? If team characteristics vary, will the scope characteristics still be valid and useful?

A useful definition of scope should stand regardless of the number of resources or duration of the project. Scope should be defined so that scope, as it exists, might be valued differently by various stakeholders. A working definition for this paper is centered on deliverables rather than process or architecture:

- **Scope is the tangible outcome of project tasks.**

Further, it is argued that scope exists (at some quality) at the conclusion of work. Scope drives cost and schedule, but is not cost and schedule. Scope is evaluated against requirements. The same scope, in different stakeholder context, may have different value. Resources may be required, but the amount of resources used does not change the scope. Scope includes intermediate deliverables necessary to achieve requirements in ultimate project output, including repeated scope due to rework.

Scope helps to clarify the boundaries of a project: what scope will be completed during the project. (Therefore, also what scope will not be included.)

2.1. The Zen of Scope

- Scope is.
- Scope isn't value, but it's existence is a basis of value.
- Scope isn't work, but is the result of work.
- It **is**.

2.2. Scope Items and their Attributes

A project's scope element as a tangible outcome of a work task will be described so that the item can be planned, designed, implemented, and evaluated. Our research on project activity modeling, starting in 1995 at the University of Tokyo, has emphasized scope as generating a demand for activity. Project models are used to forecast the realization of scope. The scope of each task is characterized by units, effort, and complexity [16] .

- **Units**. A dimension of measurement for the expected deliverable or outcome, relevant to the teams who hold domain knowledge and will be involved directly in realization of the scope. Drawings, Parts, Prototypes, Tests, Sites, Reviews, etc. These scope units are useful so that scope will be monitored. Audited. Tested. Accepted. Received. In some cases, these units of scope, by their nature (not a characteristic of project targets nor resources) are divisible during the flow of work. In other cases, the delivery of units by their nature must be as a set or in a continuous stream.

- **Effort.** Effort in nominal hours is used to describe the size of the items in comparison to a standard case. Nominal effort is not a measure of a particular team and resource, but rather a measure of the scope. See "Resource Nominal" below.
- **Complexity**. The degree of needed information, and therefore uncertainty, required to realize the scope. More complex scope requires more information to be realized. The complexity measure corresponds to the proportion of coordination activity to nominal work activity required to transfer the scope across dependencies.

Over sixteen years across many industrial workshops we have found the capture of scope in project models to be practical if the representation follows three principles: target neutral, resource nominal, and exception realistic.

2.2.1. Target Neutral

Scope items should be represented independently of any targets for the scope's realization and how the scope will be valued. For example, if a planned scope is 24 drawings, the actual scope generated is 27 drawings, these plans and actual drawings exist whether or not the target was 20 or 30 drawings. (It can be true that a plan does not meet all targets, and that actual results might not meet a plan). If project objectives change, and thus targets and how they are valued, the scope (drawings) still exists as planned and as implemented.

2.2.2. Resource Nominal

The scale of scope items can be described in comparison to a commonly held view of typical resource requirements. In other words, the scope be represented as resource nominal: given a starting point assumption of the typical, nominal team size, what is the effort of the scope?

- **Nominal** scope: a measure of the amount of output driven by **work at a common unit of resource**
  - # Stories in a day for one experienced (and uninterrupted) professional
  - # Bricks by a single average bricklayer in a day
  - Man-months

Consistent with Brooks, the assumptions of valid linearity of effort to resource availability can be stated as an inherent characteristic of the scope. Resource nominal scope is a basis of dialogue, not only for deliverables but also to anchor a description of scope and information flow across dependencies.

2.2.3. Exception Realistic

Scope should be represented so that imperfect implementation and rework can be modelled. In some cases, the failure of teams to allocate attention in a timely manner will lead to errors, and in turn the imperfect scope will be either reworked or not, in turn leading to system consequences. Therefore, the state of the scope item should include completeness and quality.

3. Scope Patterns

Scope Patterns are sets of scope items that share common characteristics including units, association to nominal effort, complexity, and exceptions. Scope patterns easily fit to the way that certain working teams think about, discuss, plan, and monitor work. These patterns are defined to meet the characteristics above: target neutral, resource nominal, and exception realistic items. These patterns allow an essential, abstract definition of the nature of the demanded work, more easily defined downstream as details of the scope are learned. Several examples of scope patterns are shown in the table below.

Table 1. Scope Pattern Examples.

Fixed Duration	The scope is demanded from the start milestone and the demand stops at a fixed duration after that start. The completeness and quality of the scope might be effected if attention and resources are not sufficiently supplied during the fixed duration. This pattern is the historical basis of most planning models and forecasts, including CPM, PERT etc.
Linear Burn Down	A demand for scope exists at the start of activity. As resources apply attention the scope is generated; the more attention and skill the faster the completion of the scope. With poor supply of attention and mistakes, the duration might extend or poor quality is accepted (if visible). Common in agile software projects.
Units non-linear Effort	Scope is described by units, yet units are not of common nominal effort. Overall progress is non-linear depending on which units are selected for attention by resources. Sometimes this non-linearity is captured by having a subcategory of units; 24 drawings, 5 of which are assembly, 10 are large parts, and 9 are small parts. Common in complex engineering projects.
Map Coverage	Scope is defined by work across a map or range; attention to an area within the range allows for completion of that portion of the scope. Feasible pathways within the map may be an inherent aspect of scope. A common pattern in infrastructure projects.
Cyclical Operations	Demand for progress is limited to a period which repeats; unattended demand may be removed, rolled over, and/or a driver of poor quality and rework. Within a period (e.g. a week, a month) the progress on scope might be similar to other patterns. Common in the administrative activities of most projects.

4. Comparison of Project Performance by Scope

It is common to evaluate the performance of projects by comparing a given project to others, with differences in scope, resources, architecture, and externalities taken into account. By describing scope as resource neutral, various resourcing options can be compared both during planning, but also in comparing project of similar scope across past actual results. In this case one should differentiate clearly between nominal scope and actual scope:

- **Nominal scope**: does not count repeat (whether intentional or rework) if these could be avoided in different project scenario of same scope.
- **Actual scope**: the total output as actually generated, including poor quality, excess quality, mistakes, and repeats

5. Conclusion

This paper has presented a definition and example uses for scope in projects. Projects are considered sociotechnical systems, therefore the cost, duration, and quality of tasks are an emergent result, influenced both by the underlying characteristics of the task activity and the position of the task in the project architecture. As such, a definition of scope is presented so that these task dynamics can be modeled, simulated, and therefore the designed. Scope is the tangible outcome of tasks, represented independently from project targets and resources. So that the project scope can be considered in combination with other project elements, the scope is modeled to be target neutral, resource nominal, and exception realistic.

Acknowledgement

Colleagues working to develop and deploy *Project Design* workshops over the last sixteen years have helped to evolve this thinking about scope and its practical application in the design of complex engineering projects. Special thanks to Ralph Wood and Roland de Filippi.

References

[1] INCOSE, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. John Wiley & Sons, 2015.
[2] B. R. Moser and R. T. Wood, Complex Engineering Programs as Sociotechnical Systems, in J. Stjepandić et al. (eds.) *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, Springer International Switzerland, 2015, pp. 51–65.
[3] B. R. Moser and R. T. Wood, Design of Complex Programs as Sociotechnical Systems, in *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, Springer International Switzerland,, 2015, pp. 197–220.
[4] PMI, *A Guide to the Project Management Body of Knowledge: PMBOK Guide*. Project Management Institute, 2013.
[5] M. N. Mirza, Z. Pourzolfaghar, and M. Shahnazari, Significance of Scope in Project Success, *Procedia Technol.*, Vol. 9, 2013, pp. 722–729.
[6] J. R. Turner, *The handbook of project-based management*, vol. 92. McGraw-Hill, 2014.
[7] L. Lee-Kelley, K. Leong, and Loong, Turner's five-functions of project-based management and situational leadership in IT services projects, *Int. J. Project Management*, Vol. 21, 2003, No. 8, pp. 583–591.
[8] F. P. Brooks, *The Mythical Man-month: essays on software engineering*, Addison-Wesley Pub. Co, Reading, 1975.
[9] M. J. Lanigan, Task estimating: completion time versus team size, *Engineering Management J.*, Vol. 4, Oct. 1994, No. 5, pp. 212–218.
[10] D. Rodríguez, M. A. Sicilia, E. García, and R. Harrison, Empirical findings on team size and productivity in software development, *J. Systems and Software*, Vol. 85, 2012, No. 3, pp. 562–570.

[11] K. Holtta-Otto and C. L. Magee, Estimating factors affecting project task size in product development- an empirical study," *IEEE Transactions Engineering Management*, Vol. 53, Feb. 2006, No. 1, pp. 86–94.
[12] K. B. Clark, Project Scope and Project Performance: The Effect of Parts Strategy and Supplier Involvement on Product Development, *Management Science*, Vol. 35, 1989, No. 10, pp. 1247–1263.
[13] C.-S. Cho and G. E. G. Jr, Building Project Scope Definition Using Project Definition Rating Index, *J. Archit. Eng.*, Vol. 7, 2001, No. 4, pp. 115–125.
[14] P. R. Dumont, G. E. G. Jr, and J. R. Fish, Scope Management Using Project Definition Rating Index, *J. Manag. Eng.*, Vol. 13, 1997, No. 5, pp. 54–60.
[15] L. Song and S. M. AbouRizk, Quantifying Engineering Project Scope for Productivity Modeling, *J. Constr. Eng. Manag.*, Vol. 131, 2005, No. 3, pp. 360–367.
[16] B. R. Moser, *The Design of Global Work: Simulation of Performance Including Unexpected Impacts of Coordination across Project Architecture*, PhD thesis, University of Tokyo, Tokyo, Japan, 2012.