Navigating Complexity [1]

By Bob Prieto

Introduction

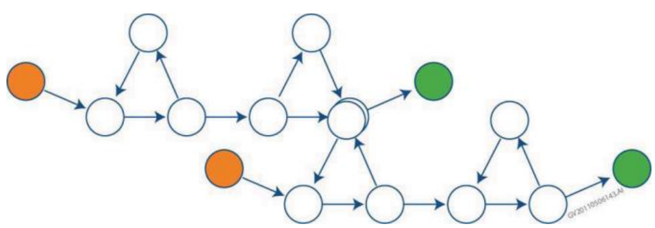
The greater the complexity of a project the greater the importance of transformational leadership. This is essential as uncertainty grows in large complex projects. A more iterative project planning process underpinned by iterative project risk assessment is required, so we may quickly understand when a project execution plan no longer maps to reality and when the project's risk profile has changed in ways that demand new strategies or reallocated focus for dealing with risk.

Large complex projects must have execution resiliency built into project plans, and frequent risk assessments coupled with complexity assessment can be used to continually seek ways to reduce complexity, strengthening resiliency in the process.

In this paper we look at cyclomatic and quantum approaches to assessing complexity in large complex programs considering analogous approaches in other domains . The first of these methods is well-established while quantum complexity has not been previously extended to the project management domain.

Cyclomatic Complexity

Every project execution plan has a number of paths that run through it. The analog is a large computer program with 100,000-plus lines of code. Cyclomatic complexity analysis allows different execution approaches to be compared from a complexity standpoint guiding us to reduced complexity approaches with a reduction in the attendant but often unseen risks.



Formal methodologies and tools, developed for use in the software and other industries, exist to characterize the complexity of a system, represented here by the program's execution plan; determine the risk exposure of such dynamic systems; and determine where the point of critical

Cyclomatic Complexity: Cyclomatic complexity (or conditional complexity) is a metric originally developed for use in the software industry to indicate the complexity of a program. It directly measures the number of linearly independent paths through a program.

[1] How to cite this paper: Prieto, R. (2024). Navigating Complexity, *PM World Journal*, Vol. XIII, Issue VI, June

complexity is and thus how close to fragility and vulnerability the proposed execution plan may be[2].

In this paper we will look at the application of cyclomatic complexity analysis to large complex projects with the objective of:

- Calculating a complexity score for a given execution strategy
- Enabling comparison of alternative execution strategies from a complexity perspective
- Assessing how complexity changes over the execution of the project to identify regions of increased complexity to enable complexity reduction
- Creating a complexity measure that is integrated with other project dashboard elements
- Strategies for reducing complexity
- Providing a model for refined complexity measures drawing on the ideas presented in "Measurement of Complexity in Large Complex Projects"[3] and the "Quantum Project Management" model[4]

Cyclomatic Complexity Analysis

Cyclomatic complexity is a metric used to assess the complexity of a program. It quantifies the number of decision points within the code and number of linearly independent paths. It reflects how intricate your program's control flow is. More paths mean higher complexity. Several well-established methodologies exist for calculating cyclomatic complexity and its use over the years has resulted in identifying strategies for reducing program complexity. These strategies provide analogs to evaluate in the context of large complex projects.

Cyclomatic complexity is computed using the **Control Flow Graph (CFG)** of the program. In the CFG, nodes represent groups of commands, and directed edges connect nodes if the second command might immediately follow the first one. Major deliverables become high-level nodes. Sub-deliverables and tasks form the lower-level nodes.

Creating a control flow graph (CFG) from a work breakdown structure (WBS) involves translating the hierarchical decomposition of tasks in the WBS into a graphical representation, such as what we may see in a Network Diagram or Gantt chart, which captures the flow of execution. Let us break down the process step by step:

[2] The GIGA Factor; Program Management in the Engineering & Construction Industry; CMAA; ISBN 978-1-938014-99-4; 2011

[3] Prieto, R. (2024). Measurement of Complexity in Large Complex Projects, *PM World Journal,* Vol. XIII, Issue IV, April 2024. https://pmworldlibrary.net/wp-content/uploads/2024/04/pmwj140-Apr2024-Prieto-Measurement-of-Complexity-in-Large-Complex-Projects.pdf

[4] Prieto, R. (2024). Quantum Project Management, PM World Journal, Vol. XIII, Issue I, January 2024. https://pmworldlibrary.net/wp-content/uploads/2024/01/pmwj137-Jan2024-Prieto-Quantum-Project-Management-.pdf

1. **Understand the WBS**:
   - Begin by thoroughly understanding the WBS. The WBS represents the project's hierarchical breakdown into tasks, subtasks, and work packages.
   - Each element in the WBS corresponds to a specific task or activity.
2. **Identify Task Dependencies**:
   - Review the WBS to identify task dependencies. These dependencies determine the order in which tasks must be executed.
   - Look for relationships such as "Task B depends on Task A" or "Task C can start only after Task D finishes."
3. **Identify Basic Blocks**:
   - In a CFG, basic blocks are essential. These are sequences of statements that have a single entry point and a single exit point.
   - Map each task or work package from the WBS to a basic block in the CFG. Each basic block represents a specific task or activity.
4. **Define Control Flow/ Network Diagram**:
   - Determine how tasks in the WBS are related. For example, if Task B depends on Task A, there should be a control flow edge from the basic block representing Task A to the one representing Task B.
   - Use arrows or edges to connect the basic blocks based on task dependencies.
   - Consider any lag time (delays) between tasks.
   - A network diagram visually represents task dependencies. You can use software tools like Microsoft Project, Primavera P6, or even draw it manually. This becomes the basis for the CFG. In Primavera P6, the Gantt chart view displays the WBS alongside the project schedule.
5. **Include Decision Points**:
   - If your WBS includes decision points (e.g., conditional branches), represent them in the CFG.
   - Create branches in the CFG by connecting basic blocks with conditional edges (e.g., "if" statements).

- Handling decision points/conditional tasks in a WBS is discussed below.

6. **Exit Points**:

- Ensure that each basic block has a clear exit point. This could be a return statement, a jump to another block, or the end of the program.
- The last basic block in the CFG should represent the final output or completion of the project.

7. **Identifying Critical Path**:

- In project management, CFGs help identify the critical path—the sequence of tasks that determine the project's overall duration.
- Focusing on critical tasks ensures efficient resource allocation and timely project completion.

8. **Validate the CFG**:

- Double-check that the CFG accurately reflects the task dependencies from the WBS.
- Confirm that there are no missing connections or incorrect sequencing.
- As the project progresses, update the control flow graph to reflect any changes (e.g., delays, scope adjustments, new tasks).

Remember that a well-constructed control flow graph helps project managers visualize the project's flow, identify critical tasks, and manage resources effectively[5] [6].

Conditional Tasks in a WBS

While a WBS does not directly handle conditionals, you can incorporate them indirectly:

1. Define separate tasks for different scenarios.
2. Use dependencies (e.g., finish-to-start relationships) to model conditions.
3. If Task A must complete before Task B, set that dependency.
4. Adjust the WBS as conditions change during the project.

---

[5] CS153: Compilers Lecture 17: Control Flow Graph and Data Flow Analysis

[6] Control Data Flow Graph - Cornell University

A WBS focuses on breaking down work, assigning responsibilities, and tracking progress. For more complex conditional logic, consider using project management software or tools that allow dynamic task dependencies.

Calculating a complexity score for a given execution strategy

Cyclomatic complexity is a metric used to assess the complexity of a project. It quantifies the number of decision points within the project and the number of linearly independent paths. It also considers the complexity introduced by connected components in the CFG. These are separate subgraphs within the overall graph in computer programs or can be thought of as modules or pre-assembled activities conducted in parallel with the main project activities for an engineering and construction project. Each of these subgraphs are the CFG for the respective component.

Cyclomatic complexity is calculated based on the control flow graph of the program. This graph visually represents the program's execution flow, including loops, conditions, and other control structures. The graph consists of nodes (representing individual steps, tasks, or WBS elements) and edges or arrows (indicating the order in which these steps are executed).

To perform cyclomatic complexity analysis for an engineering and construction project, you need the following input parameters:

- **WBS:** Specific WBS you want to analyze.
- **Control Flow Graph (CFG):** Constructed from the WBS, the CFG visually depicts the program's control flow.
- **Edges (E):** Count the number of edges in the CFG. Each edge represents a transition from one node to another. Nodes may connect to more than one other node (e.g., a loop or conditional statement).
- **Nodes (N):** Count the number of nodes in the CFG. Nodes correspond to individual steps or instructions in the WBS.
- **Connected Components (P)**: Identify the number of connected components in the CFG. These are separate subgraphs within the overall graph in computer programming or can be thought of as independent modules or significant pre-assemblies, each with their own CFG. In a CFG with no connected components P = 1, representing the overall CFG.

Cyclomatic complexity (M) is calculated as $M = E - N + 2P$

Higher cyclomatic complexity indicates more decision points and potential paths through the project. Projects with high cyclomatic complexity are more challenging to navigate and maintain on schedule. The risks of out-of-sequence work and logic errors grow with complexity.

Enabling comparison of alternative execution strategies from a complexity perspective

Complexity, in addition to scale, is a major challenge for large, complex projects and a major factor in why this class of projects experiences exceptionally high and persistent failure rates. The ability to assess complexity for alternative execution approaches allows clients and contractors to score the relative complexity of alternative approaches. It is important to underscore this complexity comparison is far from being the sole factor in evaluating alternative approaches, but it is one currently missing from alternative analysis as it relates to project execution strategies and even alternative project configurations as described by varied WBS elements.

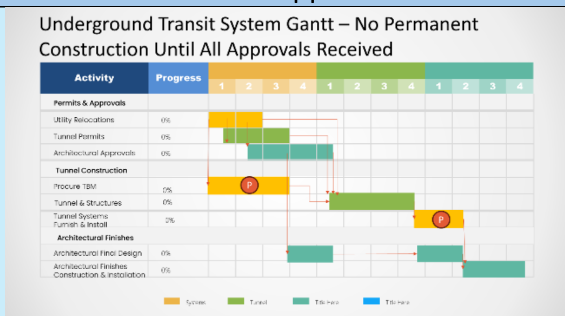
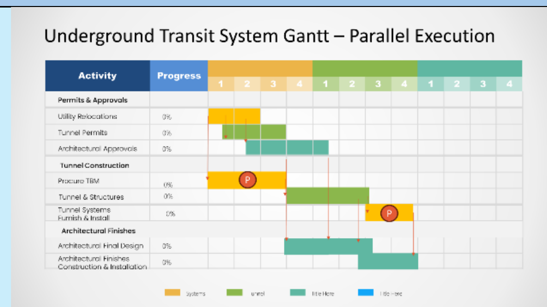
We have previously seen that cyclomatic complexity as currently conceived and employed will not account for the hidden couplings and other entanglements that are observable in the quantum like behaviors we witness in large complex projects[7].

Let us look at a simplified comparison of two project execution scenarios for an underground urban transit system. In the first scenario, the approach was to not undertake any permanent construction until all approvals, including architectural approvals often the subject of stakeholder derived delays, was received. This created a long string of precedences and extended the overall project schedule.

In the second scenario, the tunnel, systems and architectural activities were decoupled and progressed in parallel. Precedences along the critical path were reduced. In both scenarios there were two connected components consisting of extensive pre-assembly.

The Gantt charts of each planned execution strategy are shown together with a calculation of the associated cyclomatic complexity. When other factors related to schedule reduction and procurement risk were considered, scenario 2 with lower complexity was confirmed as the better execution approach.

[7] Prieto, R. (2024). Measurement of Complexity in Large Complex Projects, *PM World Journal,* Vol. XII, Issue IV, April 2024. https://pmworldlibrary.net/wp-content/uploads/2024/04/pmwj140-Apr2024-Prieto-Measurement-of-Complexity-in-Large-Complex-Projects.pdf

<table>
<tr><th>Scenario 1
Underground Transit System – No Permanent Construction Until All Approvals Received</th><th>Scenario 2
Underground Transit System – Parallel Execution</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Complexity
Underground Transit System – No Permanent Construction Until All Approvals Received
• E • 12
• N • 9
• P • 2
• M = E-N + 2P • 7</td><td>Complexity
Underground Transit System – Parallel Execution
• E • 9
• N • 8
• P • 2
• M = E-N + 2P • 5</td></tr>
</table>

If the two execution strategies involve different connected components, the cyclomatic complexity (M) of each of the connected components should be calculated and an overall value for M calculated. Similarly, if the other nodes and edges are reconfigured in the various alternatives new values for M should be calculated such that:

$M_{\text{Scenario 1}} = M_{\text{WBS}} + M_{\text{Module 1}} + M_{\text{Module 2}}$

$M_{\text{Scenario 2}} = M_{\text{WBS'}} + M_{\text{Module 1'}} + M_{\text{Module 2'}}$

Assessing how complexity changes over time

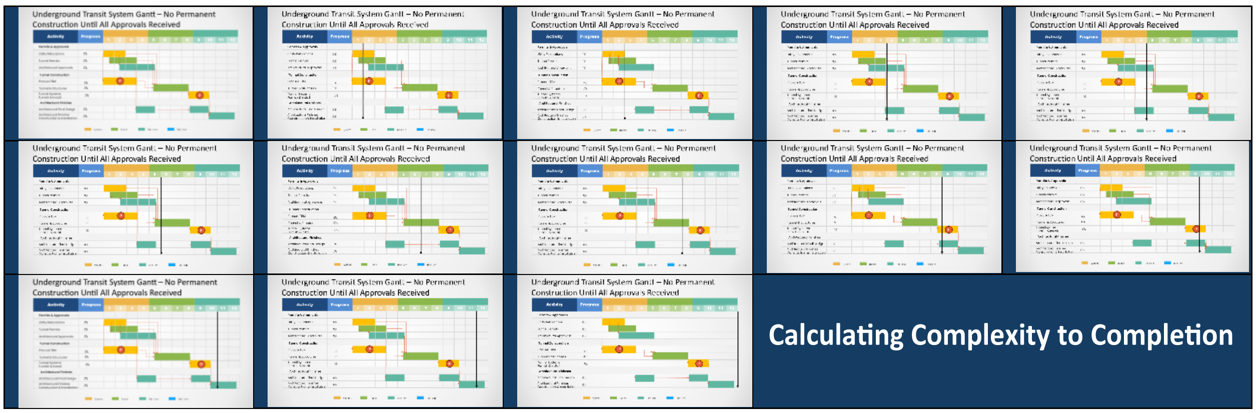
Assessing how complexity changes over the execution of the project allows the project team to identify regions of increased complexity to enable complexity reduction and increase management focus as may be required. In the first example we look only at:

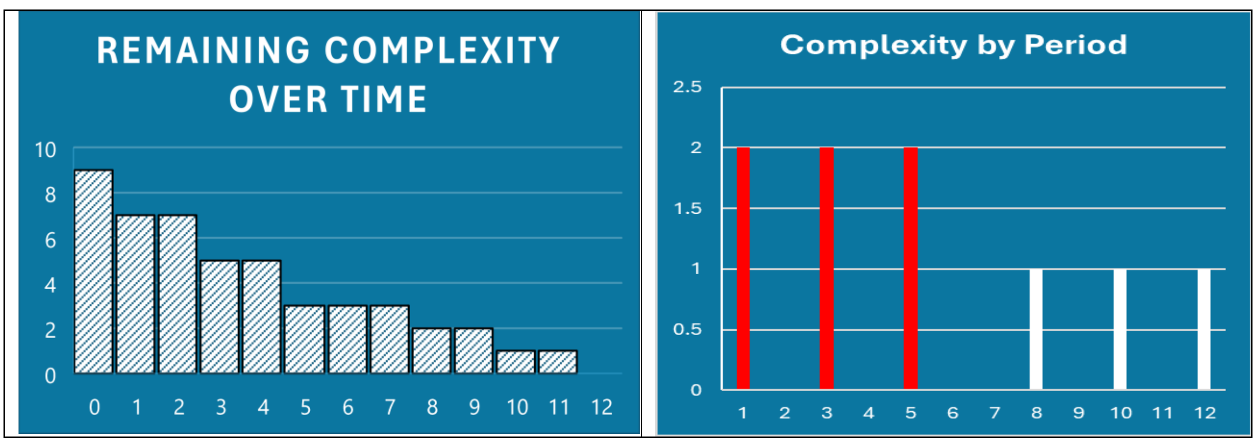
$M_{\text{Scenario 1}} = M_{\text{WBS}}$

Stepping through the originally developed WBS we see complexity decrease as the project progresses. Note the drops in complexity when each of the modules are completed. A more realistic view would be seen when we include the complexity of each of the modules and arrive at a total project complexity of:

$M_{Scenario\ 1} = M_{WBS} + M_{Module\ 1} + M_{Module\ 2}$

Similarly, the WBS used in this example is very simplistic and a more detailed and complicated WBS would give an even stronger picture of the changes in complexity. As can be seen below, periods of high complexity can be identified, and appropriate management resources and actions taken to mitigate any effects from complexity.





Creating a dashboard complexity metric

In the example in the previous section, we saw a WBS with a cyclomatic complexity score of 7. The low complexity did not consider contributions from either of the connected components and a more appropriate measure of overall project complexity would include summing these complexities with that of the other activities in the WBS. Clearly each of the connected components are likely to involve higher complexity than what is suggested just considering the WBS.

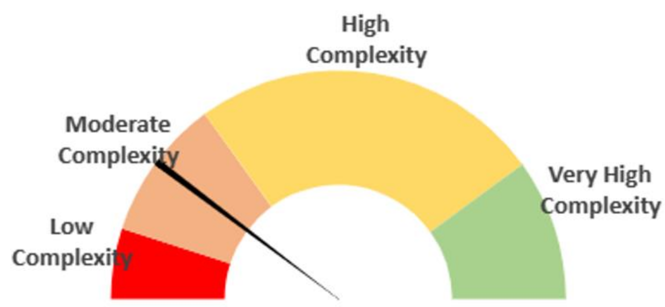
We can categorize overall project complexity through a series of complexity ranges drawing from experience in other large complex execution networks. This categorization can be summarized as follows:

1. **Low Complexity (1-10)**:
    - Projects with low cyclomatic complexity (typically between 1 and 10) are straightforward and easy to understand. We see this in the simple example in this paper.
    - These projects have minimal branching and decision points.
    - They facilitate clarity of communication and management.
2. **Moderate Complexity (11-20)**:
    - Projects in this range have moderate complexity.
    - They may contain some connected components including loops, conditionals, and branching. Attention to complexity inside connected components, such as modules or extensive pre-assemblies, warrants special attention.
    - These projects are still manageable, but project managers need to pay attention to any perturbations in execution.
3. **High Complexity (21-50)**:
    - High cyclomatic complexity indicates more intricate and interconnected project execution.
    - Projects with many decision (and hold) points, nested loops (contributing to intricate execution), and complex conditionals fall into this category.
    - These projects are riskier from a project execution and management perspective and more prone to perturbations, white space risks, and impacts of uncertainty.
    - High complexity makes effective review of the execution plan and attendant risks harder.
4. **Very High Complexity (50+)**:
    - Extremely complex projects fall into this range.
    - They are characterized by a WBS with numerous paths, deeply nested structures, and often convoluted logic.
    - They are difficult to understand, plan, simplify execution and remain on track.
    - These WBS benefit from selective disaggregation, reduction in precedences, alternative contracting strategies and increased use of more manageable and less complex modules.

Integrating complexity with other project dashboard elements

Cyclomatic complexity is a valuable discussion point during project planning and execution reviews. While cyclomatic complexity provides insights, it is just one aspect of project execution plan quality. Combine it with other metrics and good project management practices for robust project execution.

An example of a complexity dashboard element is shown below.



Other complexity measures[8] can also be provided including look aheads of upcoming complexity to inform project management. Individual complexity gauges can be provided for significant connected components especially those that lie along the critical path.

Strategies for reducing complexity

Limiting and managing complexity is an essential ingredient of achieving project success. While there are a wide range of strategies applicable in large complex project environments, there are a subset focused more on the structure of the WBS and the associate execution plan. As such there are certain strategies available to the project planner and manager that relate directly to these items. These WBS focused strategies for reducing complexity include:

- Break down (decompose) large tasks into smaller, more manageable ones. Each task or function should do one thing and do it well. Further decomposition will tend to increase complexity.
- Replace complex conditional logic with well-defined strategy patterns. Strategy patterns represent repeatable "algorithms" or ways to perform certain functions or activities where the object being acted on is interchangeable. Think of this as an optimized, standard work process such as building formwork or assembling scaffolding.
- Extract common groups of tasks into their own functions or methods. Evaluate whether the grouping is sufficiently repetitive to consider some form of standardization, pre-assembly or even complete modularization. Said another way, does the creation of connected components encourage reuse of work/process templates and support further optimization. Does it make sense?

[8] Prieto (2017) Complexity in Large Engineering & Construction Programs; PM World Journal; Vol. VI, Issue XI – November. https://pmworldlibrary.net/wp-content/uploads/2017/11/pmwj64-Nov2017-Prieto-complexity-in-large-engineering-construction-programs.pdf

- Minimize nested or repetitive activities that result in revisiting tasks and activities previously performed or completed.
- Manage complexity by developing and providing a set of proven solutions to common problems. They are templates for how to solve a problem in a way that has been proven to work.
- WBS elements should have single-point responsibility. This does not obviate the need for teamwork but provides clarity on required capabilities and accountability.
- WBS elements should not be forced to depend upon interfaces that they do not use. Stated another way, only the minimum number of required precedences should influence the timing and activities performed within a given WBS element. WBS elements should be decoupled from precedences to the maximum extent possible.
- Interfaces should be closely inspected and monitored for implied or inherited conditions, requirement, constraints or unintended or unnecessary couplings.

Sources of complexity in large engineering and construction projects are delineated in the National Academy of Construction Executive Insight, "Defining Project Complexity and Its Sources[9]"

Refined Complexity Measurement – Quantum Complexity

Cyclomatic complexity provides a first order measure of complexity utilizing a project's WBS. As projects become larger and more complex, this measure will tend to understate the actual complexity we witness and experience. The ideas presented in "Measurement of Complexity in Large Complex Projects"[10] and "Quantum Project Management"[11] provide a model for refined complexity measures.

These papers explored the application of quantum concepts to large complex projects, highlighting the interconnected nature of project elements, uncertainties, and emergent properties. Several metrics have been proposed to quantify the complexity of quantum systems and warrant consideration in the context of large complex projects. Some of these metrics include:

- **Entanglement entropy:** A significant consideration in large complex project performance especially with respect to the presence of significant and often unnecessary precedences.

---

[9] (1) (PDF) Defining Project Complexity and Its Sources Key Points (researchgate.net)
[10] Prieto, R. (2024). Measurement of Complexity in Large Complex Projects, *PM World Journal*, Vol. XII, Issue IV, April 2024. https://pmworldlibrary.net/wp-content/uploads/2024/04/pmwj140-Apr2024-Prieto-Measurement-of-Complexity-in-Large-Complex-Projects.pdf
[11] Prieto, R. (2024). Quantum Project Management, PM World Journal, Vol. XII, Issue I, January 2024. https://pmworldlibrary.net/wp-content/uploads/2024/01/pmwj137-Jan2024-Prieto-Quantum-Project-Management-.pdf

- **Quantum circuit complexity:** This can be thought of as the number of transformational processes and steps required in LCP as well as the "length" of the critical path or other activity chains.
- **Quantum Fisher information:** This measures the sensitivity to small perturbations, providing insights into the complexity of the tasks and an assessment of the fragility of the execution plans.

Entanglement Entropy

Entanglement entropy in large complex projects is focused on the identification of significant and often unnecessary precedences. In Scenario 1, previously described in this paper, no permanent construction was undertaken until all approvals were received. The execution strategy delayed the start of the tunnel and tunnel systems while the project waited for architectural approvals. In this instance these approvals were an unnecessary precedence. Scenario 2 showed the result of eliminating this unnecessary precedence. Scenarios 1 & 2 were based on real life examples where the shift to the Scenario 2 approach necessitated a change in contracting strategy and a reassessment of risk.

Precedences can be counted, for example, in Primavera P6 using the Predecessors column in the Activities view. If an activity has multiple predecessors they need to be counted individually. Insight into the precedences in Scenario 1 informed the changed execution approach in Scenario 2. Understanding precedences provides a logic check on the WBS and supports optimization of the project schedule.

The overall number of precedences speaks to the complexity of the execution program but sheer numbers will be misleading if the overall scale of the project is not considered. One possible metric would be to calculate the ratio of the number of precedences to WBS tasks. Think of this as Precedence Density.

Precedence Density = Precedences/WBS elements

Similarly, this complexity measure can be looked at over time to see how this metric changes as the project evolves. In this instance the forward looking Precedence Metric would be calculated as the ratio of number of precedence remaining to number of WBS elements remaining.

This check of complexity can show periods of higher complexity. Consider the situation on a large process plant where many WBS activities are related to site excavation, grading, trenching and large mat continuous pours of concrete. The number of precedences in this early phase of the project are few. But as we approach a later piping or cabling stage, the density of precedences associated with those WBS activities may be very high as materials receipt, cleaning, installation sequence, welding and weld inspection create many interfaces both within and between various systems.

Identification of unnecessary predecessors or open-end activities can be accomplished in Primavera P6 by filtering activities for no Predecessors and similarly filtering for no Successors.

Combining the two lists will show activities without predecessors or successors. A logic check of the schedule will show missing predecessors or successors.

In our Scenario 1 example architectural permits has no successor related to tunnel or tunnel systems and is therefore an unnecessary precedence.

Quantum circuit complexity

Quantum circuit complexity can be thought of as the number of transformational processes and steps required in a large complex project as well as the "length" of the critical path or other activity chains. Scheduling features in Primavera P6 identify the critical path as well as other paths containing activities designated as critical or having substantive impact on the critical path. One can then display the activities in each path as well as the order in which they are undertaken. Retaining the logic in the P6 schedule, including early and late dates, allows handling of completed activities and therefore an ability to "look ahead" through the balance of the schedule.

Quantum circuit complexity for a large complex project is best calculated as a Level 4 schedule (can be calculated at other levels but complexity assessment is weaker for Levels 1,2,3) and consists of:

- Number of Level 4 activities (A)
- Number of Critical activities in the WBS (C) or near critical flow paths impacting the critical path
- Number of activities on the critical path (L)

Schedule Levels	
Level 1 Schedule (L1 Schedule)	Executive Summary or Project Master Schedule (PMS). The Level 1 schedule is represented as a single bar summarizing project data
Level 2 Schedule (L2 Schedule)	Management Summary or Summary Master Schedule (SMS). The Level 2 schedule expands on Level 1 information and shows activities and major items of capital equipment
Level 3 Schedule (L3 Schedule)	Project Coordination Schedule (PCS) or Publication Schedule. The Level 3 schedule focuses on specific project coordination activities
Level 4 Schedule (L4 Schedule)	Further breakdown of activities from the Level 3 schedule providing more detailed information on specific tasks and activities.

Schedule Levels	
	The Level 4 schedule includes specific task details and dependencies
Level 5 Schedule (L5 Schedule)	Detailed Schedule The Level 5 schedule provides the most detailed view of project activities

A quantum circuit complexity (QCC) metric can be constructed as a function of these three measures such that:

QCC = f(A, C, L)

QCC consists of:

- A **scale factor** (A/50,000) where 50,000 has been assumed to be a nominal WBS activity level for a Level 4 schedule for a large complex project. Smaller WBS are less complex while higher ones are more complex. The ratio is shown here as having linear scaling of complexity but more likely a Power Law relationship exists.
- A **complexity factor** (C/A) where C is all critical activities within the WBS including those on the critical path and A is the count of all activities. An alternative assessment of complexity may be to consider the number of near critical paths impacting the critical path (including the critical path itself) divided by all flow paths in the WBS. A schedule with too few critical activities may lack detail or missing logic. One with too many critical activities may be overly aggressive.
- A **degree of difficulty factor** (L/20) where 20 has been assumed to be the nominal number of WBS activities along the critical path.

QCC can be written as:

QCC = f(scale factor, complexity factor, degree of difficulty factor)

Quantum circuit complexity often involves a combination of both multiplicative and additive factors. A quantum algorithm may depend on the product of gate complexities (multiplicative) and the sum of qubit complexities (additive).

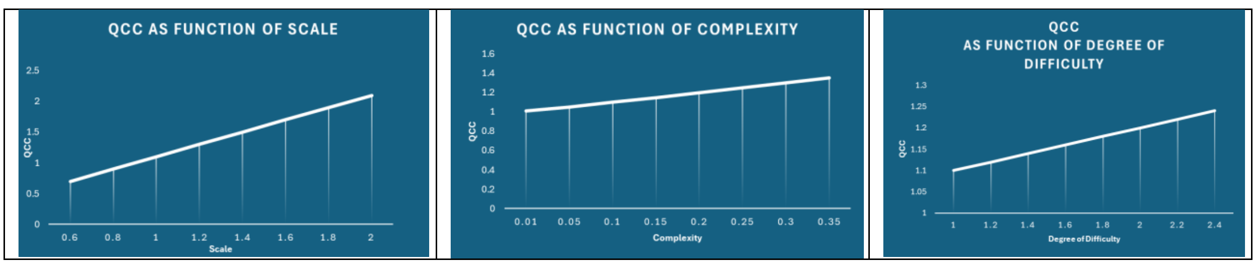
Quantum gates are analogous to specific operations or transformations applied to qubits. Think of project management actions as the analogs of quantum gates. Just as quantum gates modify the state of qubits, project management actions (e.g., task assignments, resource allocations, **dependencies**) impact the progress and state of a project.

Qubits are the fundamental units of quantum information and can exist in superpositions of states and exhibit quantum entanglement. In the context of a P6 schedule, consider project tasks as the analogs of qubits. Each task represents a unit of work or activity within the project. Like qubits, tasks can be in various states (e.g., not started, in progress, completed) and may have dependencies (entanglements) with other tasks.

Using this as an analog we might state the product of the gate factors (project management actions) as being proportional to the degree of difficulty times the complexity factor from above. This is turn would be summed with the scale factor above to calculate Quantum Computational Complexity. We can express this as:

QCC = (Degree of Difficulty x Complexity) + Scale

This calculated Quantum Computational Complexity holds unless the overall project network behaves non-linearly or even exponentially as described below. As such QCC as calculated above should be regarded as a minimum or threshold value and tends to behave linearly as can be seen in the pairwise relationships below.



The scalability of complexity in a network does not follow a strict linear or exponential pattern. Instead, it depends on various factors, and the relationship can be more nuanced.

In some cases, complexity may grow linearly with network size. For example:

- **Adding Nodes**: If you simply add more nodes to an existing network, the complexity may increase linearly. Each new node introduces additional configuration, management, and potential interactions.
- **Simple Topologies**: In small, straightforward networks with minimal interconnections, the complexity might scale linearly.

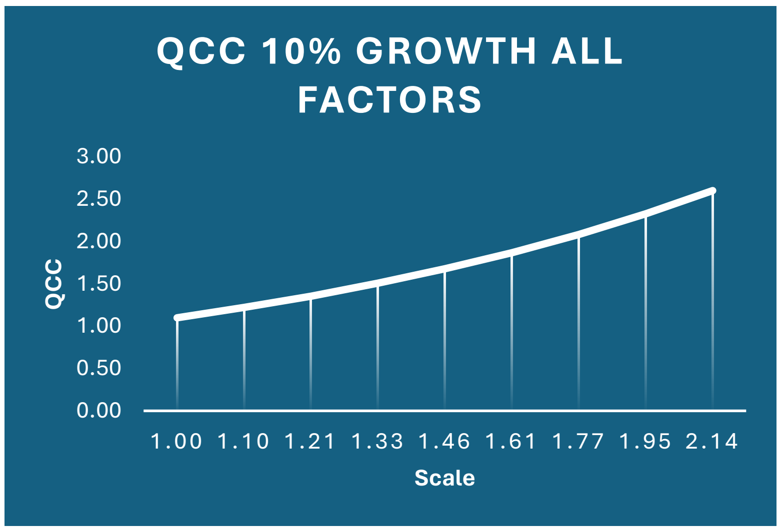
Complexity often exhibits exponential growth due to interactions and dependencies:

- **Interconnections:** As nodes connect to each other, the number of potential interactions increases exponentially. Each new link introduces additional paths for work and information to flow, leading to complexity.
- **Emergent Behavior:** Complex systems exhibit emergent behavior—unpredictable outcomes arising from interactions between components. These emergent properties contribute to exponential complexity.

- **Configuration Dependencies:** Configurations (physical and management processes) can have dependencies on each other. As the network grows, these dependencies multiply, leading to exponential complexity.

Real-world networks often exhibit nonlinear complexity growth:

- **Hybrid Models:** Some aspects may grow linearly (e.g., adding nodes) while others grow exponentially (e.g., interactions; edges in the control flow graph (CFG) grow faster than nodes).



- **Threshold Effects:** Complexity might remain manageable until a certain point, after which it suddenly escalates due to critical mass or emergent behavior.
- **Feedback Loops:** Feedback loops can amplify complexity, leading to nonlinear growth.

Quantum Fisher information

There is not a direct analog to Quantum Fisher information but there are several insights we can gain by studying it.

Quantum Fisher information measures the sensitivity to small perturbations, providing insights into the complexity of the tasks and an assessment of the fragility of the execution plans.

Schedule Comparison

This allows us to assess the potential impacts of small perturbations in a project execution plan. Built in capabilities in Primavera P6 allows you to compare two versions of a schedule (e.g., an updated version vs. a baseline) to identify differences in various schedule variables. Just as quantum Fisher information provides insights into the precision of parameter estimation in quantum systems, the Schedule Comparison helps identify changes and discrepancies in project schedules.

Comparing Schedule Variables

When performing a schedule comparison in P6, you can compare variables such as duration, relationships, constraints, and other schedule elements. Consider this as examining different observables (variables) in a quantum system. Each variable contributes to the overall behavior of the system. Like how Quantum Fisher information characterizes the sensitivity of quantum states to parameter variations, comparing schedule variables helps assess the impact of changes on project timelines.

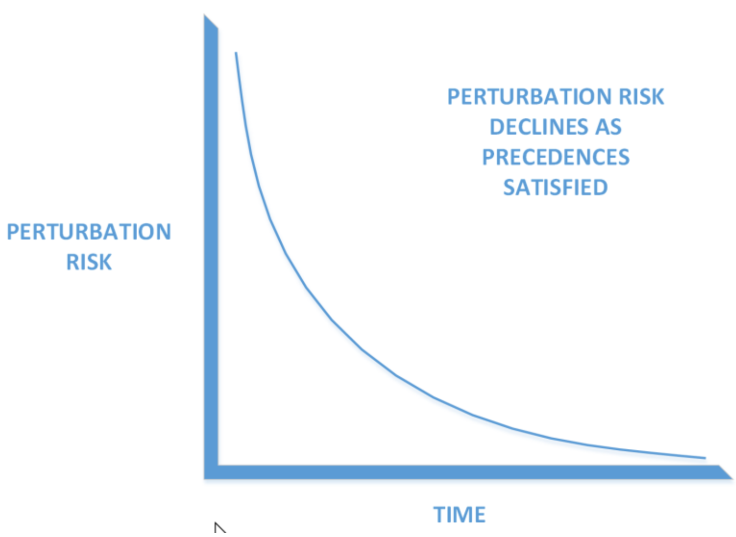
Detecting Changes

Schedule comparison tools highlight differences between schedules, making it easier to spot modifications. In quantum systems, detecting changes (e.g., due to noise or **perturbations**) is crucial for maintaining stability and accuracy. Detecting changes ensures that project managers can address discrepancies promptly and maintain project alignment.

While there is not a direct one-to-one correspondence, the schedule comparison serves a similar purpose by analyzing differences and providing insights into schedule variations.

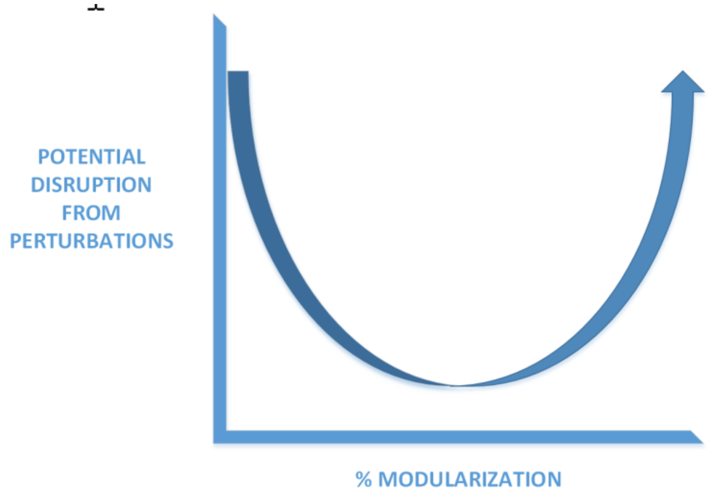
A word on perturbations

Previously[7], we have looked at the risks to large complex projects that may derive from perturbations. In that work we saw that perturbation risk declines as precedence are satisfied.

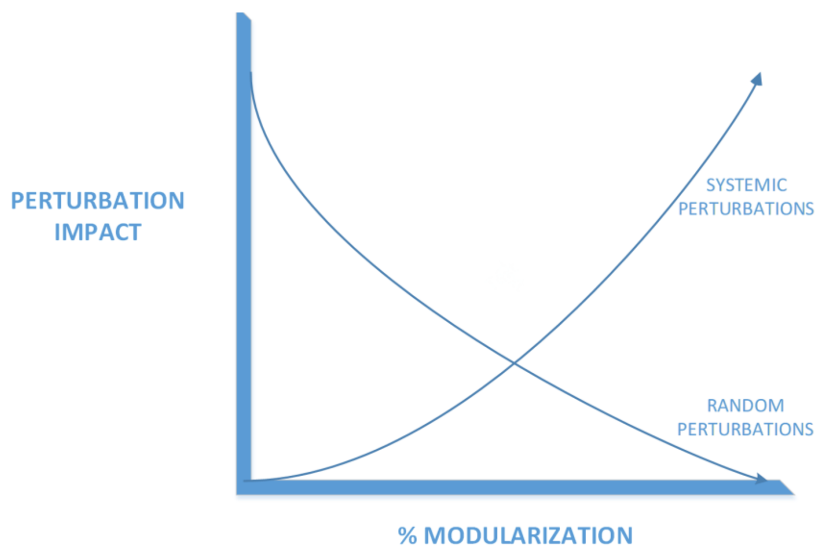


Also, in that work we looked at the effects of modularization on the impacts from perturbations. We noted the relationship between complexity and modularity of "systems" and that Nature's complex systems reward modularity and its ability to limit the effects of perturbations while at the same time recognizing that excessive modularity exposes the system negatively to the effects of even stronger perturbations.

We see from natural and other complex system analogs that the potential disruption from perturbations declines with modularization, reaching some minimum potential after which further increases in modularization increase the level of disruptions from perturbations.



This behavior in the impacts of perturbations can be understood by considering two different classes of perturbations. The first is associated with purely random discrete events. As we increase the modularization of the system, susceptibility to these random perturbations are more localized and contained. Conversely as we increase the modularization of the system susceptibility of the system to perturbations more systemic in nature grows. In the second class of perturbations second and third order couplings, including coupling through constraints, grows in scope and importance. We can see the behavior of each of these classes of perturbations below.



Conclusion

This paper provides an exploration of complexity assessment in large complex projects, drawing parallels with analogous approaches in various domains. By examining the application of cyclomatic and quantum methodologies to assessment of complexity in the management of large complex projects, this paper illuminates crucial strategies for effectively managing complexity and uncertainty. Key insights highlighted include the significance of transformational leadership in navigating uncertainties, the benefits of utilizing cyclomatic complexity analysis to streamline project execution plans, and the importance of integrating complexity assessment with other project dashboard elements for resilient project execution. Stressing the importance of iterative planning, regular risk assessments, and complexity reduction strategies, this resource empowers project managers with the necessary tools and insights to enhance project resiliency and achieve success in the realm of large complex projects. By embracing these principles and implementing best practices, project professionals can optimize outcomes, mitigate risks, and confidently navigate the complexities inherent in project management.

References

Prieto, R. (2024). Measurement of Complexity in Large Complex Projects*, PM World Journal,* Vol. XIII, Issue IV, April 2024. https://pmworldlibrary.net/wp-content/uploads/2024/04/pmwj140-Apr2024-Prieto-Measurement-of-Complexity-in-Large-Complex-Projects.pdf

Prieto, R. (2024). Quantum Project Management, PM World Journal, Vol. XIII, Issue I, January 2024. https://pmworldlibrary.net/wp-content/uploads/2024/01/pmwj137-Jan2024-Prieto-Quantum-Project-Management-.pdf

Prieto (2017) Complexity in Large Engineering & Construction Programs; PM World Journal; Vol. VI, Issue XI – November. https://pmworldlibrary.net/wp-content/uploads/2017/11/pmwj64-Nov2017-Prieto-complexity-in-large-engineering-construction-programs.pdf

CS153: Compilers Lecture 17: Control Flow Graph and Data Flow Analysis

Control Data Flow Graph - Cornell University

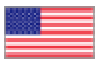
Defining Project Complexity and Its Sources Key Points (researchgate.net)

The GIGA Factor; Program Management in the Engineering & Construction Industry; CMAA; ISBN 978-1-938014-99-4; 2011

About the Author

Bob Prieto

Chairman & CEO
Strategic Program Management LLC
Jupiter, Florida, USA



Bob Prieto is Chairman & CEO of Strategic Program Management LLC focused on strengthening engineering and construction organizations and improving capital efficiency in large capital construction programs. Previously Bob was a senior vice president of Fluor focused on the development, delivery and turnaround of large, complex projects worldwide across the firm's business lines, and Chairman of Parsons Brinckerhoff.

Bob's board level experience includes Parsons Brinckerhoff (Chairman); Cardno (ASX listed; Non-executive director); Mott MacDonald (Independent Member of the Shareholders Committe); and Dar al Riyadh Group (current)

Bob consults with owners of large, complex capital asset programs in the development of programmatic delivery strategies encompassing planning, engineering, procurement, construction, financing, and enterprise asset management. He has assisted engineering and construction organizations improve their strategy and execution and has served as an executive coach to a new CEO.

He is author of nine books, 950 papers and National Academy of Construction Executive Insights, and an inventor on 4 issued patents.

Bob's industry involvement includes ASCE Industry Leaders Council, National Academy of Construction and Fellow of the Construction Management Association of America (CMAA). He serves on the New York University Abu Dhabi Engineering International Advisory Council and previously served as a trustee of Polytechnic University and the Millennium Challenge Corporation Advisory Board. He was appointed as an honorary global advisor for the PM World Journal and Library.

Bob served until 2006 as one of three U.S. presidential appointees to the Asia Pacific Economic Cooperation (APEC) Business Advisory Council (ABAC). He chaired the World Economic Forum's Engineering & Construction Governors and co-chaired the infrastructure task force in New York after 9/11.

He can be contacted at rpstrategic@comcast.net.