NEXTGEN AGILE METHODS - AGILE ROLLING WAVE PLANNING

Paul W.M. Cuypers

Hyves Project Research, Næroset, Norway

ABSTRACT

Project planning is subject to diminishing returns and contradictory demands. While management requires a comprehensive overview of activities, costs, and benefits, detailed planning for an uncertain future can be wasteful, as reality quickly catches up with the plan. Agile rolling wave planning is a multi-level, multi-horizon planning approach designed to address this challenge. This paper outlines the theoretical foundation and practical application of agile rolling wave planning during a ServiceNow implementation project. A systems thinking model of project uncertainty is presented, along with four fundamental counter strategies. Based on the model, it is demonstrated how agile rolling wave planning reduces uncertainty by providing relevant information and facilitates horizontal coordination between teams. A roadmap for AI-enabled project management software to support the method is outlined. Agile rolling wave planning techniques enable organizations to transition from risk to uncertainty management, applying agile principles beyond the software development realm.

KEYWORDS

Next-generation agile, project success, agile rolling wave planning, scrum, uncertainty management, AI.

1. INTRODUCTION

Although the introduction of agile methods in project management marked a significant advancement, it is not without challenges. In traditional agile methods, the incremental and iterative approach to managing the product backlog results in the project scope evolving, making it difficult to define in advance what management will receive for their investment. Another challenge is scaling and coordinating actions and decisions among multiple agile teams in complex projects. Scrum teams are intentionally kept small, and with a flexible scope, the need for coordination is addressed vertically through the Scrum-of-Scrums principle. This creates a hierarchical chain of command, which is not particularly known for enabling swift and timely decisions and can lead to delays and cost overruns.

A known challenge in traditional agile is its application in project domains beyond software development. The key principle of welcoming new requirements through an iterative and incremental approach stems from the almost limitless ability of software to accept new components, a condition typically absent in industrial projects. For instance, adding twenty extra floors to an office building on a whim isn't feasible if the foundation and lower structure cannot support the extra weight. In software development, new code can be generated by reusing old code, unlike tangible project deliverables where materials may end up being scrapped. Industrial projects typically receive only one chance to succeed while staying within the allocated timeline and budget.

David C. Wyld et al. (Eds): SESBC, AIFL, NLPTT – 2025
pp. 01-17, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151501

Another advantage of software development is that the deliverables are constant. Each development cycle must produce functional software, a list of known issues, and release notes. In industrial projects, the deliverables depend on the design choices made. For instance, when constructing a zero-emission house, the choice of energy source determines whether the project requires solar panels, a wind turbine, or both. Changes in deliverables may also involve engaging with new partners or vendors who adhere to different compliance regulations, operating procedures, roles, responsibilities, and skills for the parent organization. From a theoretical standpoint, classic agile is an empirical method grounded in a set of prescriptive principles and practices. Although it is effective in software development, the underlying mechanisms or active ingredients remain uncertain. Without a theoretical and normative foundation, a free-for-all culture leads to various agile implementations, often mixed with traditional Prince2 or PMBOK, making it challenging to determine the relationship between project method and project success.

In addition to the inherent challenges of agile methods, there are structural hurdles facing project management, both as a practice and a science. First, the lack of a universal definition of project success and the failure to establish a comprehensive project methodology concept. A unified definition of project success is a prerequisite for demonstrating causation between the project method and the likelihood of success. Despite being the largest area of research, the concept of project success remains elusive. After fifty years of research, a consensus has emerged that project success is a subjective, multidimensional construct dependent on time, context, and stakeholders [1][2], essentially an elaborate way of saying 'it all depends.' A clear definition of success is critical; without one, the effectiveness of project management in enhancing the chances of project success cannot be established.

Another factor complicating the proof of causation is the failure to differentiate between method and methodology. Project management methods, such as Agile and Prince2, are often referred to as methodologies instead of methods. Methodology systematically examines the effectiveness, efficiency, limitations, and applications of methods. The critical methodology question in research is: What if another method had been used? Could we have reached a different conclusion? The analogous methodology question in projects is: If another project management method had been used, could we have achieved better results? Project management methodology, or the meta method for systematically studying the effectiveness and efficiency of project management methods, is essential for knowledge development and for creating a comprehensive research agenda, which includes the development of AI technology.

Without a comprehensive methodology, the project management community sees the primary challenge as effectively integrating agile and waterfall approaches, rather than aligning theory with practice. This study introduces a theoretical model of project uncertainty and a practical method to mitigate uncertainty by making information accessible through systematic planning. Planning is a continuous effort throughout the project and often regarded as a necessary evil. While planning requires time, it also saves time; any time invested in planning reduces the hours available for producing results. Furthermore, planning experiences diminishing returns: as planning stretches further into an uncertain future, it becomes less likely to remain realistic and beneficial. When plans confront reality, a growing gap between the plan and actuality necessitates a replanning effort.

Conflicting interests exist: The parent organization management requires comprehensive scope, benefit, and cost projections based on detailed plans. However, the team tends to postpone detailed planning until the latest responsible moment to capitalize on emerging insights. The research question is: How can we balance the need for accurate overall project information with the necessity of leveraging emerging information to create relevant short-term detailed plans?

This research introduces agile rolling wave planning as a multi-level and multi-horizon planning approach designed to maximize the use of emerging information and increase the chances of project success.

The paper is structured as follows:

- Literature review
- A conceptual model of project uncertainty is presented
- Agile rolling wave planning principles and techniques are outlined
- Case observations of a practical application of agile rolling planning during a ServiceNow implementation project are reviewed
- A roadmap for project management software to support the method is presented.

2. LITERATURE REVIEW

Whitney et al. [3] describe the need for rolling wave planning in complex IT projects, as the standard approach of decomposing a phenomenon into ever-smaller elements fails to exert more control over those elements, resulting in an ineffective understanding of the system's behavior.

Treating decomposed hierarchies as static modules increases the risk of technical incompatibility. Hence, the need for rolling wave planning, described as a form of progressive elaboration commonly used in agile methodologies.

Rolling wave planning is defined as initially high-level planning in the Work Breakdown Structure (WBS) that evolves into more detailed planning on an iterative basis. They observe that emergence invalidates the notion of modularity for complex projects, yet decomposition remains a common practice in large coding teams.

The rolling wave planning process, as described by Githens [4], utilizes a Work Breakdown Structure (WBS) based on phases applicable to new product development, information systems, and other technical development environments. The method is primarily suited for inventive assignments. The branches of the WBS are developed incrementally after approval, captured by 'time buckets' or planning horizons.

In relation to projects and dynamic environments, Collyer & Warren [5] state that high levels of control are said to inhibit the adaptability needed to maximize business benefits in dynamic environments. In the rolling wave approach, the plan for each phase is finalized at the end of the preceding phase, enabling improved environmental adaptation.

Giammalvo [6] describes the application of rolling wave planning in the construction industry where there are multiple external delaying factors, like weather, equipment breakdowns, material delays, and the effects of COVID on the work capacity, as a series of time horizons or look ahead schedule, ranging from 7, 30, 90, 180, and 365 days.

Hanan & Ford [7] define two approaches in an adaptive project environment: Rolling Wave and Progressive Elaboration. A Rolling Wave lifecycle utilizes an iterative planning process (Progressive Elaboration) to capture increasingly detailed information regarding scope, schedule, cost, risk, and quality as the project progresses.

The PMBOK 6 (Project Management Body of Knowledge) [8] defines rolling wave planning, as an iterative planning technique where work in the near term is planned in detail, while work further in the future is planned at a higher level. This allows for flexibility and adaptation as more information becomes available and the project progresses.

3. METHODOLOGY

The research methodology involves developing a theory using a systems thinking approach and conducting case observations by applying the derived practical techniques in the field. One limitation is that the observer is part of the team, providing an inside-out perspective, while an outside-in viewpoint could be a more desirable holistic alternative. Additionally, mixed roles that involve both research and program management increase the risk of bias and subjectivity. Given the small sample size and the complexity of factors affecting project execution, positive observations do not necessarily prove causation between the chosen method and project success.

4. PROJECT UNCERTAINTY

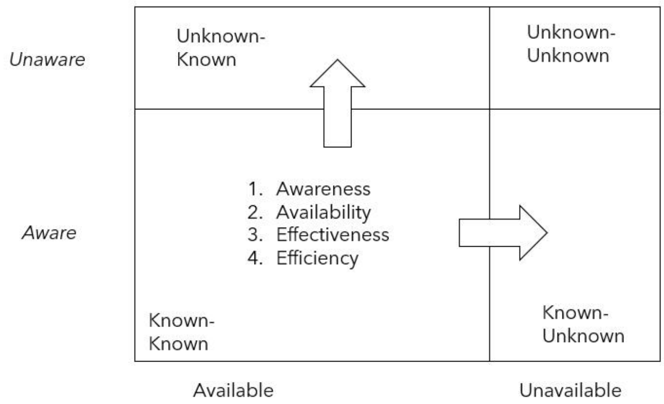

Figure 1. The Uncertainty Matrix and Uncertainty Reduction Strategies.

A project is a temporary organization tasked with completing a specific assignment within constraints like time and budget while maximizing benefits based on the investment. To achieve success, control is necessary over relevant project dimensions, including requirements, stakeholders, resources, deliverables, and activities. Regardless of the chosen control dimension, the question arises: what information is unavailable or what is the team unaware of? From a meta-level perspective, achieving project control implies mastering uncertainty. Uncertainty can be defined as a lack of awareness or availability of information, forming a matrix with the quadrants [9]:

– Known-known refers to available information, and its relevance is understood
– Known-unknown: the information is needed but not available
– Unknown-known, relevant information exists, but one is unaware
– Unknown-unknown relates to unavailable and unaware information.

Having a conceptual model of uncertainty, the next step is to define project uncertainty.

A project can be modeled as a system with elements and relations, for example: Assignment, context, decisions, method, and scenarios. A *system* is defined as a whole that contains two or more parts, each of which potentially affects the whole's properties or behavior. None of the parts has an independent effect; how any part affects the whole depends on what others are doing. The elements of a system are always connected; between any parts, there is always a direct and an indirect path. The systems thinking approach makes sense of the real-world complexities by looking at the whole rather than splitting them into parts compared to functional and process analysis approaches.

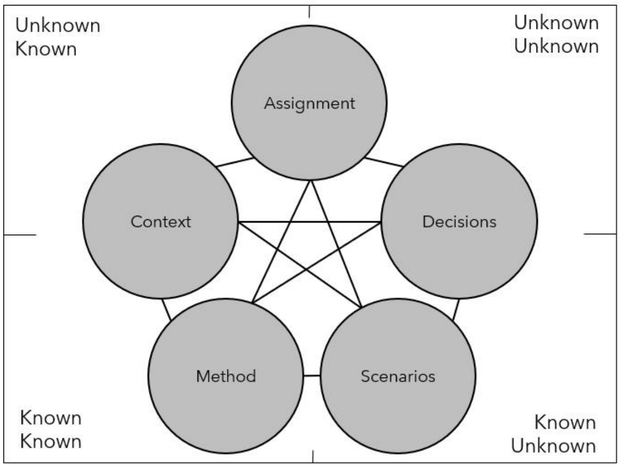

Figure 2. A systems thinking model of project uncertainty [9].

Projecting the uncertainty matrix onto the system creates a conceptual model of project uncertainty. Each quadrant of the uncertainty matrix applies to:

1. The system's elements
2. The interactions between the elements, and
3. The effects of interventions on the system's states.

In theory, when all relevant information is known and applied, there is no uncertainty. Elemental strategies to reduce uncertainty are:

1. Raising information awareness
2. Increase information availability
3. Improve the effective use of information
4. Maximize information efficiency.

Raising awareness of information and improving availability aim to maximize the known-known area. Information effectiveness refers to the ability to formulate accurate plans, solve problems, and make informed decisions using relevant information. Information efficiency pertains to the time and resources needed to gather pertinent information. Agile rolling wave planning reduces uncertainty by enhancing information awareness, improving information effectiveness, and maximizing efficiency.

5. AGILE ROLLING WAVE PLANNING

5.1. Principles

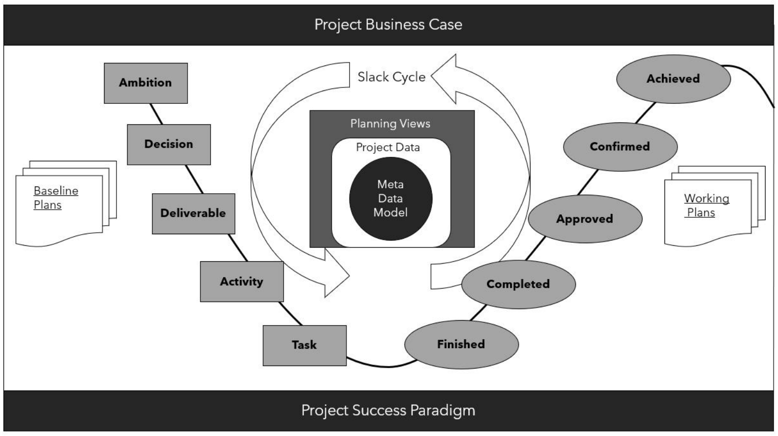

Figure 3. Agile rolling wave planning overview.

Agile rolling wave planning is based on planning what is known, proactively eliciting information, providing for future scenarios, and making in-flight corrections using emerging information. Information is gathered proactively based on the project metadata model, defining relevant information necessary for planning, decision-making, and execution. The project is planned in logical waves of ambition, with each wave broken down into planning at different levels of aggregation, characterized by increasingly shorter time horizons and greater detail down to the task level (figure 3). Two sets of plans are maintained: the baseline version, as approved by the steering committee, and a working version. The business case serves as the top-level document guiding the project, stating what success looks like. The success paradigm addresses whether success has been achieved. The following paragraphs outline the main elements of the method.

5.2. Project Metadata Model

The project metadata technique is based on the work of psychologists Chabris and Simons [10], where viewers are asked to perform a simple task, such as counting the number of passes a basketball team makes in a video. A person in a gorilla suit passes in the middle of the scene. Surprisingly, most viewers remain unaware, focusing on the job. *Selective attention*, or inattentional blindness, is the price we pay for concentrating on a task. In the daily fog of project meetings and looming deadlines, the team is prone to overlook essential information, such as prerequisites, dependencies, problems, solutions, or necessary decisions, which can often be hidden in plain sight. Biases are another complicating factor; as one desperately wants the plan to work; ignoring contradicting information is not an uncommon phenomenon.

When viewers are warned beforehand, most subjects notice the gorilla and complete the task correctly. The Project Metadata Technique (PMDT) is based on the principle that constructing a metadata model of essential information increases information awareness while performing other activities. The premise is that any emerging information can be mapped to a discrete set of metadata elements, capturing crucial project details. Each metadata element has a set of required attributes. By completing the element’s attribute list, more information becomes accessible, reducing uncertainty.

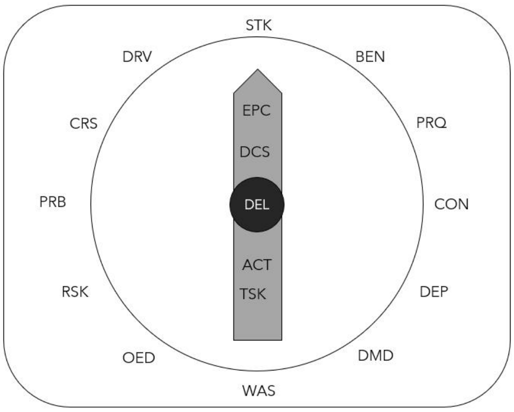

Figure 4. The Compass project metadata model.

The *Compass project metadata model* (figure 4) covers the following information elements:

1. North, driver (DRV), stakeholder (STK), benefit (BFT)
2. East, prerequisite (PRQ), constraint (CON), dependency
3. South, demand (DMD), working assumption (WAS), open end (OED)
4. West, risk (RSK), problem (PRB), and crisis (CRS)
5. Compass needle, epic (EPC), decision (DCS), deliverable (DEL), activity (ACT), and task (TSK).

The cardinal directions represent contextual factors, while the compass needle describes the assignment. The premise of PMDT is that any uncovered information can be translated into a metadata element, such as meetings, emails, or conversations, by asking the question: What is the practical implication of this information for the project? More information is elicited by defining attributes. A benefit has a name, description, unit of measurement, a baseline value measured at project start, and a target value.

A decision includes a description, its nature (urgent vs. important), reach (global or local), alternative choices, and an owner. When crucial, the latest point in time for making the decision without delaying the project must be determined. Confirmation is essential: how and when can the project ensure that the decision is correct and reverse the action if necessary? A project driver is typically defined by a name, description, and classification (risk, problem, opportunity, threat). Keeping track of drivers is vital; after the project is completed and the dust has settled, all drivers must be adequately addressed. The stakeholder metadata element encompasses a role (project owner, manager, business analyst, architect, developer, tester, or client), belongs to a team (test, architect, or developer team), and falls under an organization (parent organization, partner, consultancy, supplier), while also having a physical location. This information helps keep track of time zones and meeting opportunities.

A prerequisite includes a description, an owner responsible for implementing it, a deadline for when the prerequisite must be met to avoid delaying the project's critical path, and a status. A constraint consists of a description and a qualification (hard or soft). Common project constraints include budget at completion, contractual end dates, laws, mandatory policies, and industry standards that must be followed. Dependency is defined by a description and a qualification, either hard or soft. Working assumptions are consciously made hypotheses or estimates. They are not facts; some value or approximate range must be provided to proceed with the creation of business cases or project plans in the absence of accurate information. Typical working assumptions pertain to resource needs, activity lead times, work hours, and the (financial) benefits the project will deliver. A working assumption has the following attributes: a description, a verification method, the first opportunity to check the assumption, and an owner responsible for conducting the verification and notifying its accuracy.

A demand governs an activity and includes a description, an owner, and its implementation status. For example, welding entails a series of safety measures, such as wearing a helmet and gloves, and having a fire extinguisher nearby. Note the difference between a requirement and a demand; wearing a helmet shapes the activity, not the result. An open-ended movie or book refers to films and books where the author or director does not fully conclude the plot, leaving questions unanswered and prompting viewers to consider what happens next. In the project context, an open end refers to known-unknowns. Open ends are a subtype of a task, an assignment to discover missing information, characterized by a question, an answer, an owner, and a deadline indicating when this information is needed.

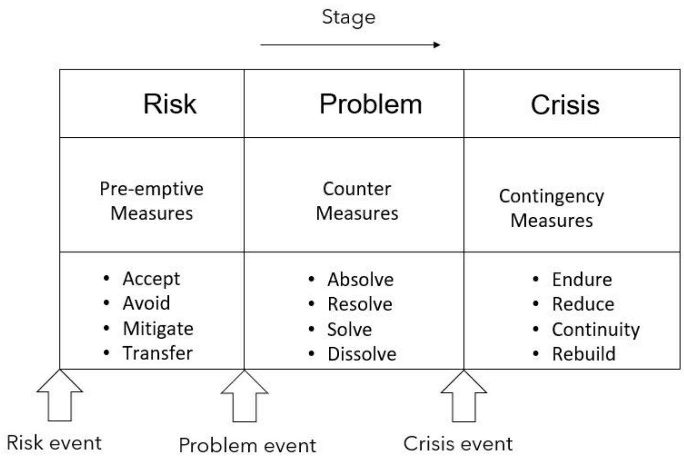

Figure 5. The risk continuum and counter strategies.

A risk is a possible event that causes a problem threatening project success and is characterized by a probability, the likelihood of detecting that a problem event has occurred, first order effects (impact), second order effects (recovery), and third order effects (collateral damage). A problem is defined as the gap between the intended or mandatory state and the actual state, characterized by a description, nature (urgent vs important), owner, and solutions. A crisis occurs when all attempts to solve the problem have failed, leaving the only option as reducing the damage.

A risk event occurs when risk is introduced into a situation; project initiation converts theoretical risk into actual risks, such as project failure. When a risk is identified, a fundamental decision must be made whether to remain passive or to become proactive by investing project time and resources in defining and implementing pre-emptive, counter, and contingency measures (figure 5). Note that adverse events do not solely cause problems; any contextual factors, such as prerequisites, constraints, and dependencies, that are either overlooked or not effectively addressed will lead to issues, including elements of the assignment, such as poor decisions, unidentified deliverables, or unclear requirements.

5.3. Planning levels and views

To improve situational awareness, threat assessment, direction, and coordination, project information is continuously gathered, following the SLACK cycle principle: Scan, log, assess, change, and keep. An assessment evaluates the impact of emerging information, determines which parts of the plans must change, and identifies what can be retained. The information is captured in plans at different levels of aggregation using various planning views:

1. Calendar plan
2. Gantt chart
3. Network plan
4. List view
5. Kanban Board
6. Result Breakdown Structure.

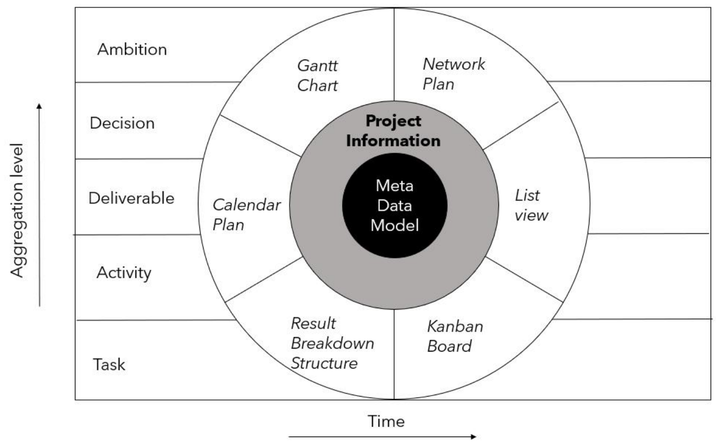

Figure 6. Planning views and aggregation levels.

The top-level plan represents the ambition level, where the complete project timeline is outlined in epics. An epic is a visionary statement that captures success, such as a product, service, capability, or establishing an opinion. Since epics are high-level statements, they are typically planned using ballpark periods in a calendar plan, such as epics per month. Alternatively, a network plan at the epic level can visualize the dependencies between epics. The network defines sequential paths, where completing one epic depends on achieving a previous epic or a parallel path, representing logically or technically independent achievements.

The next planning level is the decision level. Achieving an epic typically requires design choices, which in turn determine the deliverables. Decisions are minimally managed using a decision log. Urgent decisions are prioritized first, while important decisions are sequenced according to the latest point in time they must be made to avoid delaying the project. A decision network visualizes the logical dependencies between decisions and independent choices. Based on the cost of the alternatives, the cheapest and most expensive paths through the network can be calculated.

The deliverable level is typically managed using a network plan that shows the logical dependencies between deliverables, or a Result Breakdown Structure (RBS). Note the difference between the project result and the project outcome. In the context of project management, the result refers to the sum of all deliverables, while the outcome is the degree to which the business case is realized.

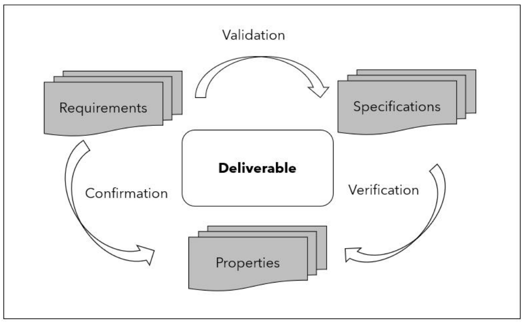

Figure 7. The attributes and test types related to project deliverables.

The RBS is a deliverable-oriented breakdown of the project result, featuring top-level phases and associated epics. Tracking deliverables is key; according to the agile principle, progress is measured against approved deliverables. Deliverables are both tangible and intangible project results characterized by requirements, specifications, and properties, and are subject to approval using a fit criterion to assess fitness for purpose (figure 7). Requirements are captured in either use cases, user stories, or free text. Compared to classic agile approaches like Scrum, where epics are broken down into requirements (stories), the intermediate decisions and deliverable levels make agile rolling wave planning a generic method applicable beyond the software development domain.

Using an RBS, the optional stages containing epics are divided into logical groups of decisions, deliverables, and other factors such as prerequisites, dependencies, constraints, risks, and issues. The RBS typically consists of nested groups; for instance, in a zero-emission house, the renewable energy epic includes the decision between alternative solar panels and a wind turbine option. The breakdown characterizes each alternative in groups with related prerequisites, constraints, risks, and vendor choices. Selecting one alternative may prompt new decisions, resulting in a breakdown and visualization of logical information groups.

The RBS is instrumental in calculating project work hours and costs using bottom-up estimates based on single-point or PERT estimates, as well as in tracking progress. When multiple alternatives are available, a working assumption is made, and the recommended alternative is incorporated into the budget. Additionally, the least and most expensive branches can be calculated to establish the limits of the ballpark figure for work hours and costs. Absolute and relative estimation in story points, commonly utilized in Scrum, can be combined using the hour-to-point conversion rate.

Note the difference between a Work Breakdown Structure (WBS) and a Result Breakdown Structure (RBS). An RBS is deliverable-oriented and consists of nouns, while a traditional WBS is activity-oriented and focuses on verbs. The RBS will include test reports, not testing. From a deliverable perspective, the Test Report end deliverable breaks down into sub-deliverables, such as test scenarios, test data, test systems, and test results. A key principle is to select the highest level of aggregation to achieve the desired degree of control and minimize project management overhead. Experts and professionals typically do not need activity or task-level planning to support their work; insight into the epic (the why) and deliverable level (the what) is often sufficient, rather than the how outlined in activity and task plans.

Deliverables are created through activities. Typical activity level plans are Gantt or PERT charts used to calculate project lead time, identify critical path activities, and manage resource scheduling and leveling. Note that the RBS sub-plan was created before the activity plan; project activities are a means to an end, not an end in themselves. A cross-check is necessary to ensure that a main activity exists for each top-level deliverable. The lowest level of aggregation is task-level planning, which typically involves using task lists or Kanban boards. A task is defined as a job for a single person. Most projects require specialist work and often involve working alone, frequently across multiple project assignments. A per-person task-based plan is necessary to optimize their time and priorities.

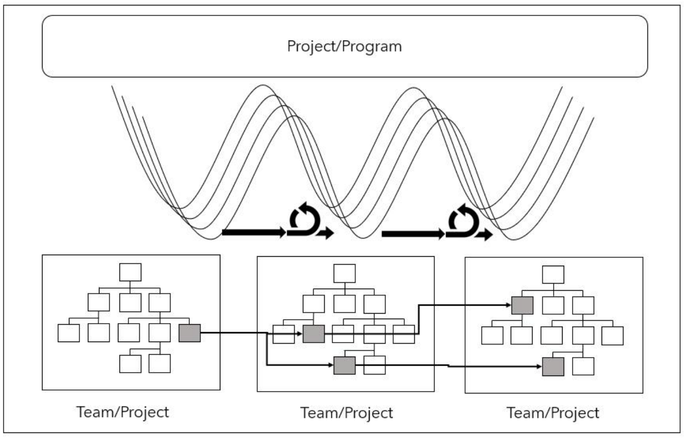

Figure 8. Coordination across teams or projects.

The classic rolling wave principle involves dividing work and segregating it over time. Projects within a program or teams within a project focus on epics and deliverables selected for maximum independence and minimal need for coordination. Although they may seem independent, dependencies can arise at lower levels, either logically or due to resource constraints. Design choices made by one team can have local or global effects; a seemingly logical decision at one stage can turn out to be a liability for decisions made in the next. Network plans across teams at various levels of aggregation provide direction and coordination among teams working on different assignments in a wave or across parallel waves from different teams.

5.3. The Project Success Paradigm

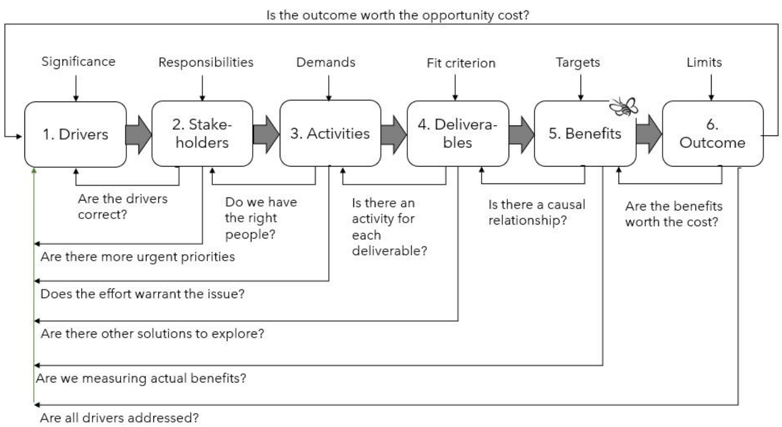

Figure 9. The project success paradigm.

Although defining project success is one of the largest areas of research in project management, scientists disagree on a definitive answer. However, there is a consensus that the iron triangle definition of success is not an accurate indicator. The correct result, on time and within budget, remains the most common yardstick used by project owners and managers in the field. The project owner, sponsor, and steering committee must continuously evaluate the status and forecasts to make strategic decisions to:

1. Proceed as planned
2. Re-plan
3. Put on hold
4. Terminate.

The project success paradigm assesses project success across all stages of the project life cycle. The feed-forward cycle encompasses project drivers that engage stakeholders to implement activities, generating deliverables that ultimately yield benefits to achieve the project's success. First-, second-, and third-order feedback is provided to ensure consistency and compare the targets for each element with the actual results. If any questions raise doubt, there is a need for in-flight corrections at one or more levels of the rolling wave planning. The numerous loops emphasize the complexity of defining and evaluating project success. Note that these are two separate questions. To achieve complete and continuous success, the answers to the model’s questions must be meaningful to any stakeholder over time. In zero-sum game assignments, where the benefits of one stakeholder group come at the expense of others, this is unattainable.

6. CASE: IMPLEMENTING SERVICE NOW ITOM & ITSM

The application of agile rolling wave planning was studied during the implementation of ServiceNow in the IT Operations Management (ITOM) and IT Service Management (ITSM) domains at Orange Business Services (OBS), an IT service provider with locations in Norway, Sweden, and Germany. OBS provides data center, workspace, sovereign, and public cloud-based services specifically to high-security customers who require total data control, reliability, and redundancy. ServiceNow, now becoming the de facto standard, was the primary driver of the program, alongside the replacement of legacy systems.

The main challenge of the assignment was adapting the system to best match the processes while leveraging new technological opportunities. Another consideration was the application's complexity; design choices made in one area can significantly impact effectiveness and efficiency in others. As a core system, the replacement influenced all employees and customers across various business areas, necessitating 10 integrations with other systems. The program comprised 17 projects, with distributed teams working across multiple European countries. Over three years, two consulting firms offered vital knowledge and increased capacity to the team. Besides introducing new systems and processes, achieving a significant cultural and mindset shift was crucial. Early in the program, a strategic decision was made to invest significantly in project management training for the core. As an IT company, the focus on technological knowledge is abundant; aligning technical expertise with next-generation agile methods fosters self-reliant teams that require minimal coordination and deliver high output.

7. DISCUSSION

7.1. Literature

The literature review agrees that planning is an ongoing process throughout the project, highlighting two key concepts: rolling wave planning and progressive elaboration.
Progressive elaboration involves developing more detailed plans based on emerging information and insights. Although these concepts are frequently mentioned in various publications, practical implementation and application information are limited, while definitions remain ambiguous. One source of ambiguity is the subjective nature of definitions, such as short-, medium-, and long-term planning. The concepts of waves, stages, and phases appear to be used interchangeably, along with increment and iteration.

The Hanan & Ford definition [7] suffers from a circular structure: *Two approaches in an adaptive project environment are Rolling Wave and Progressive Elaboration. A Rolling Wave lifecycle utilizes an iterative planning process (Progressive Elaboration) to capture increasingly detailed information.*

In the literature sample, rolling wave planning is primarily synonymous with a staged planning approach, where detailed planning for the next stage is deferred until the next decision gateway is passed. This indicates that the deliverables of the previous stage must be successfully achieved to move forward with the project and justify the allocation of resources for further detailed planning. This encapsulates the essence of the stage-gate transition technique, along with a divide-and-conquer strategy. Based on the descriptions, progressive elaboration seems to align with iterations, while rolling wave planning is incremental, which is, strictly speaking, inaccurate. In the context of this paper:

1. *Iteration* is defined as redoing tasks until the aim is achieved
2. *Elaboration* as adding additional tasks necessary to achieve the initial aim
3. *Incremental* is used to add another aim.

In the uncertainty matrix, the information availability line will eventually shift to the right as time passes and new information emerges. The question arises regarding whether emerging insights are based on a proactive or reactive approach. A passive attitude involves following the plan and adjusting as necessary when new information becomes available. A proactive approach consists of eliciting information and updating accordingly. In Scrum, proactivity is integrated through practices and ceremonies. Emerging information about the domain is gathered based on user feedback from the demo and backlog grooming and incorporated into incremental sprint planning. Retrospectives optimize the ways of working, and estimation capability is refined through product backlog re-estimation and adjusting the planned burndown rate. Inherent uncertainty reduction approaches are key components that contribute to the success of the Scrum project management method for software development.

In PMBOK v6 [8], progressive elaboration is the iterative process of increasing detail in a project plan as more information becomes available. Prototyping is mentioned as a technique used during the Collect Requirements process, providing a tangible model to elicit stakeholder feedback and identify problems early on. Prototyping is one form of proactively increasing information availability (strategy 2); other options include:

- Hypothesis
- Forecasting

- Simulation
- Calculation
- Investigation.

In the context of project methodology, the systems model illustrates the challenges of demonstrating causation between project method and success (figure 2). Project success can be regarded as a system state where all elements and their interactions contribute. Even if the chosen method is appropriate based on the assignment's nature and context, improper implementation will lead to negative effects. Similarly, even with a correctly implemented method, poor decisions and ineffective scenario planning will also result in detrimental effects. In cases of project failure, unraveling the complex interactions among the elements that cause the failure can be challenging.

7.2. Case Observations

Emphasis was placed on project management training for the team members due to the program's three-year period, an investment that paid off. Despite being structurally understaffed by more than 50% throughout this time, a small, dedicated, and disciplined team achieved significant results. While the NextGen agile techniques initially raised eyebrows, the clarity of the benefits soon led to a critical mass that readily assimilated newcomers. With the PMDT technique, team members gradually adopted a shared language, enabling effective communication and resulting in shorter meetings despite a large audience.

The implementation was carried out using a combination of tools, including Jira, Confluence, PERT/WBS Professional, and Microsoft Teams/SharePoint. Jira was adapted to accommodate ticket types that align with the Compass project metadata model. Each of the 17 projects under the program received a Jira board that is accessible to all teams, enabling horizontal information flows between projects. Epics were assigned to projects, allowing for maximum independence while minimalizing the need for coordination.

In complex systems, dependencies are inevitable, and decisions made in one project can have far-reaching global consequences. By filtering a specific ticket type, such as decisions, team members can quickly gain an overview of ongoing decision processes in other projects and evaluate whether their team should be stakeholders at the table. Biweekly decision meetings are held, during which the decision backlog is prioritized based on urgency and importance, enhancing proactiveness and reducing delays caused by open decisions. A three-minute rule was established; if a decision could not be reached within that time, the decision owner would call for a separate meeting, thus saving time spent in meetings. Similar approaches were implemented for problem and risk categories. Overall, a shift was observed from technical meetings to management meetings, where tickets were quickly evaluated, prioritized, assigned, and addressed in smaller breakout sessions.

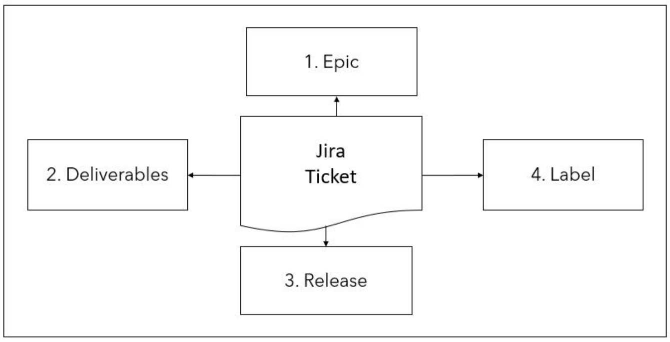

Figure 10. Linking tickets for retrieval and information coherence.

One team member noted that the chatter on Slack had decreased because information could now be found in Jira through filtering and linking. A common issue with Kanban board applications is the overwhelming number of seemingly unrelated tickets in the system, which create endless lists to scroll through in search of relevant information. To tackle this issue, each Jira ticket must, at minimum, be linked to an epic and, ideally, to a deliverable. For software, a release is indicated along with a status label such as "rejected," "on hold," or "testing." By using filters, one can quickly view all related tickets for a specific epic, deliverable, or release, including personal work lists.

Several dashboards provided insights into the project's status and progress. By filtering by ticket type, the need for a separate decision, risk, and problem log was eliminated. A decision dashboard displayed the rate of newly initiated decisions compared to those made. Typically, at initiation, the number of uncovered decisions exceeds the rate of those resolved. A turning point occurs when the rate of resolved decisions surpasses that of newly discovered ones. Although this graph can change at any moment, this pattern indicates that the team is currently proactive. A similar graph for Open Ends can be created, illustrating how known-unknown information transitions into known-knowns. This dashboard logic can also be applied to confirm Working Assumptions or estimates.

7.3. Software Support

The generic capabilities required for agile rolling wave planning are:

1. Define a project metadata model with elements and attributes
2. Detect project metadata elements
3. Targeted communication informing stakeholders of relevant information
4. Visualize the date per ambition level using a calendar, list, Kanban, Gantt chart, network chart, and result breakdown structure view
5. Prioritize metadata elements to urgency or importance
6. Determine dependencies
7. Workflow control.

Since NextGen's agile practices are experimental, tooling support remains limited. Most popular applications primarily use list-type and Kanban planning views, but they lack breakdown and network functionality. The software suite for the program included Jira, Confluence, PERT/WBS Professional, and MS Teams/SharePoint, which proved effective, as tickets in Jira could be linked to RBS items in WBS Professional and Confluence, creating makeshift integrations between systems. Another limitation is that Jira does not allow nested breakdowns for all ticket types. Ironically, progress toward better software support is hindered by the roadmap model used by most vendors; requirements and feature requests are honored only if enough users make a similar request. This allows for evolutionary improvement but hampers radical innovation.

The main requirements for an agile rolling wave planning tool are to store separate ticket types with attributes for each metadata element and to display project data in any planning view at any aggregation level (figure 6). Additionally, it should link data elements and add labels. For example, the project deliverable level can be visualized using a breakdown structure or a network plan that shows the dependencies between deliverables, similar to decisions and epics (figure 8). Note that in complex systems, elements may have sequential relationships at lower levels that are not present at higher levels of aggregation, as well as dependencies between seemingly independent team assignments.

For future development, the capabilities mentioned above can be supported by utilizing AI to assist in identifying essential project information within project communications based on the project metadata model. Once this level is achieved, the next step is to identify logical relationships among the elements, such as dependencies or groupings in a breakdown structure. Based on project roles, stakeholders can be notified about newly uncovered information that is relevant to their responsibilities. The final level would involve providing plans at various levels of aggregation in different views and assisting with resource scheduling.

8. Conclusions

Agile rolling wave is a next-generation agile method designed to reduce project uncertainty by enhancing information awareness, availability, effectiveness, and efficiency. This method addresses the challenges of scaling agile in large, complex project assignments and programs, applying agile beyond the software development domain, and reconciling the conflicting demands of long-term planning versus short-term needs. Its theoretical foundation is based on projecting the uncertainty matrix onto a systems thinking model of a project, which defines elemental strategies for reducing uncertainty. Agile rolling wave planning is an iterative, incremental, and elaborative approach. Essential project information is determined and elicited using a metadata model. The gathered information is processed at various levels of aggregation, including ambition, decision, deliverable, activity, and task levels. The time horizon at the ambition level encompasses the entire project, whereas the task level is updated daily. The project metadata model raises awareness about critical contextual and assignment-related information in any project communication. Emerging information is scanned, logged, assessed, and utilized to update the various planning views per aggregation level. A practical project success paradigm with feedback questions facilitates in-flight corrections. As an experimental method, software support is currently limited. The agile rolling wave method provides a theoretical and practical foundation for the further development of AI support in projects.

References

[1] Ika, L. A. (2009) "Project Success as a Topic in Project Management Journals: A Brief History", Project Management Journal, 40(4), 6–19

[2] Frefer, A. A., Mahmoud M., Haleema H., Almamlook R. (2018) "Overview Success Criteria and Critical Success Factors in Project Management", Industrial Engineering & Management, (7)1

[3] Kaitlynn M. Whitney et al. (2013) "The Root Cause of Failure in Complex IT Projects: Complexity Itself ", Procedia Computer Science 20 325 – 330

[4] Githens, G.D., (1998) "Rolling Wave Planning", Proceedings of the 29th Annual Project Management Institute 1998 Seminars & Symposium, Long Beach, California, USA: Papers Presented October 9 to 15, 1998
[5] S. Collyer, C.M.J. Warren (2009) "Project Management Approaches for Dynamic Environments", International Journal of Project Management 27, 355-364
[6] Giammalvo, P., (2022) PM World Journal (ISSN: 2330-4480) "The Origins and History of Earned Value Management", Vol. XI, Issue IX – September
[7] Hanan, J., & Ford, S. (2024), "An Early Warning System for Programs Utilizing the Rolling Wave Lifecycle", PM World Journal, Vol. XIII, Issue IX, September. PM World Journal (ISSN: 2330-4480)
[8] PMI (2017), A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 6th Edition, Project Management Institute, Inc., Newtown Square
[9] Cuypers, P.W.M., (2023) Mastering Project Uncertainty, A Systems Thinking Approach, Routledge; New York
[10] Chabris C, F., Simons D.J., (2010) The Invisible Gorilla: How Our Intuitions Deceive Us, New York; Random House.

ACKNOWLEDGMENTS

The author would like to thank the core project team from Orange Business Services and the embedded consultants from Einar & Partners and Devoteam; specifically: Mina Kashefi, Sigurd Våg Aaknes, Thorgeir Marthinussen, Maksym Bashkirov, Emma Lindt, Henrik Lukkarila, Hülya Patir, Besir Aliti, Håkon Nygård, Felicia Wiberg, Andrei Stanescu, Daniel Billing, Alexander Rosendahl, Alexander Ljungstrom, Alexander Karlsson, Tom Tran, Alex Lawrance, and Thor Arne Krogh.

AUTHOR

Paul Cuypers is an independent researcher, program manager, consultant, and course instructor on advanced topics such as project methodology, uncertainty management, and next generation agile techniques, working in Oslo, Norway.