Notes - Organizational Development - Development Options

- Dr Nick Hayward

A brief introduction to development concepts and options with example usage for software projects.

Contents

Intro

As a team or project lead and manager, there are various qualities and characteristics that help create a reliable system and pattern to development, its general management, and ongoing sustainability.

Such characteristics customarily evolve from experience gained during many years of project and application development, working in teams of varying sizes and structure, and learning from various role models and mentors.

However, these charateristics are not a universal set for all project leads and managers, but instead various commonalities we may perceive from successful projects. This might include clear and demonstrable communication relative to the current project, the ability to clearly articulate the planning and requirements for the project, time management to ensure the plan is met and executed efficiently, and various other skills to help the project up to and including the successful release of the end product.

Each of these general qualities need to be qualified and structured to help their application in a reliable and predictable manner from project to project, to help provide stable leadership and management. From a practical perspective, this might include clear structure and guidance for design, an application's underlying structure, and various project, department or company wide style guidelines for publishing project code, updates, and general maintenance.

It should be the responsibility of each project's lead or manager to set and define such standards, expectations, and ensure they are maintained as the project progresses from version to version. This helps all interested parties, including existing team members, potential collaborators, and any subsequent new hires and contract developers. It helps with the general onboarding process, for example, as projects and teams evolve and adapt to varying work and market conditions.

Such demonstrable skills should help foster reliable, repeatable, and professional working conditions for each managed project.

Planning

After drafting and defining a given project's plan, it's necessary to effectively and efficiently communicate its meaning and intention to interested parties.

A key part of this communication is ensuring the clarity of the plan matches expectations for deliverables and the experience and abilities of each team member with time management and deadlines. In effect, as developers we may often promise more ambitious deadlines and results than feasible in reality. Part of the

role of a project lead and manager is to ensure that reality fits the expectations of the planning, and effective pace is maintained to fit the defined plan.

Such planning may also include various external factors that need to be managed, and predicted as far as possible, by each project's lead.

Management of project

An effective project manager is a key consideration as each development project, and its associated team of developers, designers &c., is organised for a given application. Such a role becomes increasingly valuable as the team continues to add extra developers, designers, and collaborative members.

In the context of many development and design projects, a project manager helps to manage and track the defined vision for a project, and its application, to ensure it is being effectively executed and sustained. This role may also help with general management tasks, including resource allocation and maintenance, day to day oversight, task allocation where appropriate, report generation, compliance oversight, and many other associated admin and management tasks.

However, it should not be the responsibility of a project manager to solely oversee the diligence and work ethics of the project group, or individual team members. In effect, they should not be used to maintain discipline and act as a formal *task master* for the company. Such roles will, customarily, fall within the purview of line managers, department heads, &c.

Instead, a good project manager may ensure various items are being completed on schedule, keep track of issues, their timely resolution, and monitor general project output to keep it aligned with project versions, milestones, and the overall vision.

A manager should also manage news and information both in and out of the group, ensuring it is available in a timely manner for the group, in a manner that does not overly disrupt productivity. Likewise, they should coordinated and control news updates and press releases, where approriate, for the project and its application.

project manager role A few, key characteristics of this role might be considered as follows

- defining scope - a project manager should ensure projects stay on track, meet client requirements, and achieve organisational objectives. Their role involves defining scope, managing resources, promoting team communication, and mitigating risks.
- planning and scope - in the early stages of a software project, the project manager is the architect defining its initial approach relative to vision, objectives, proper scope, and deliverables
- requirements and analysis - facilitate vital stakeholder collaboration, comprehensively refining project requirements and definition commonly to ensure project stays on track and is developing according to vision
- resource management - a primary function of a project manager is to orchestrate the use of resources with precision
- leadership skills - must have strong leadership skills, be able to manage competing priorities, and should have a deep understanding of software development methodologies and tools
- communication - as communicators, a project manager should be able to bridge technical teams and stakeholders, ensuring timely, budget-friendly, quality software solutions
- risk management - identifying and managing project risks is also a part of their responsibilities
- technical knowledge - a good working knowledge of project estimation techniques and excellent technical knowledge are also beneficial for this role
- general skills - attention to detail and ability to work with many disparate sources of information, collaborators, and manage multiple, ongoing tasks…

task and time management As noted, a project manager may also be responsible for effective management and review of resources, ongoing development, and time scheduling for efficiency and various milestones.

A key option for such management in development projects is a *ticketing system*, such as services provided by Atlassian's *Jira* service management platform.

A ticketing system in software development projects is a centralised platform that helps manage, track, and resolve internal tickets, which can include tasks, bugs, and feature requests.

Specific features may vary from system to system, but expected common functionality may include the following

- task organisation - provides a central hub for organising tasks, bugs, and feature requests
- efficiency - a well-organised ticket system helps developers quickly identify the tasks they need to tackle
- transparency - everyone on the team has a clear view of the current tasks, their statuses, and who's responsible for each item
- accountability - team members can see who's working on what, making it easier to hold each other accountable
- collaboration - good ticket management facilitates better communication and collaboration among team members
- prioritisation - by managing tickets effectively, team leads can ensure that the most critical or time-sensitive tasks get priority

Common usage patterns for various roles in software development projects may be defined as follows

- project managers - may use the ticketing system to break tasks down into sprints, clearly delineate tasks for software engineering or other business divisions, track progress &c.
- developers - developers use the system to stay on top of their workloads and priorities, easily identifying the tasks they need to tackle, timescale, deadlines &c.
- testers - in a CI/CD pipeline, a log may be autogenerated each time a developer introduces a code change to the repository, and a ticket generated every time a tester finds a bug in the code change
- …

In effect, we may see how effective and key a *ticketing system* may become relative to project management in software development. Such systems, commonly, help improve communication, ranking the perceived or defined importance of a given job, thereby promoting effective project control.

Project definition and structure

The nature and scope of a project may vary from context to context, often in response to specific requirements and expectations. Use of the term *project*, itself, may be subject to debate and discussion depending upon the underlying nature and extent of the work to be completed. The work, for example, might require a few days to a few years to complete. How might this factor determine the nature or designation of a task as a project.

If we remove such temporal considerations, we may begin to consider various common traits perceived useful and important for projects. For example, a defined goal for the project.

Regardless of perceived time for a project, it should have a reason to exist and continue. Such a goal does not need to define or relate *how* something will be achieved, or the length of time required. Instead, it will commonly define the criteria to declare *success* for a project, effectively bringing the project to a close.

Such outcomes will often be useful beyond the engineering and development team, providing terms and concepts readily understood by other members of a department, company &c. These outcomes will provide transparency, and an opportunity to communicate beyond the team and project. It may also act as a usfeful initial point of reference for considering how a project is apportioning time and money for their work.

We might consider example outcomes for a project as follows,

- upgrade the import process for customer data, removing reliance on local data storage, and migrating resources and interface to a cloud-based provider…

With a project outcome initially defined, we need to then consider how we may implement the working solution and achieve the required end goal. For the context of software development, we may consider alternative methodologies to help structure and organise implementation of a project, effectively breaking it down to reach its defined final goal and outcome.

Two common methodologies include the traditional approach, *Waterfall*, and, relatively speaking, the more modern method *Agile*.

structure - waterfall The *Waterfall method* is a traditional project management methodology that is linear and sequential. It was introduced by computer scientist Winston Royce in the 1970s. The method is named *Waterfall* because each phase of the project cascades into the next, following steadily down like a waterfall.

The Waterfall method typically consists of between five and seven main stages, including

- requirements - this phase involves gathering all the project requirements, determining the project's scope, and outlining the resources required for the project
- design - in this phase, solutions that meet the requirements are developed
- implementation - this phase involves the actual creation, coding, implementation, and testing of the solution
- verification - the built solution is tested against the requirements to confirm that the project meets initial expectations
  - may commonly include a separate phase just for *testing* depending upon specifics of the underlying project
- deployment - after testing the project, it is determined it is now ready for release, including appropriate distribution, installation, and presentation to a client (where appropriate)
- maintenance - during this phase, any defects discovered or new versions of products needed are addressed by a dedicated team, including options such as patches, and ongoing fixes and issue resolution

Such stages may include other considerations, as noted, depending on context and use requirements for this model.

Perceived benefts may be considered as follows,

- clear structure - the Waterfall method uses a clear and defined set of steps to follow when compared with other methodologies
- intuitive progression - intuitive to follow and adopt, commonly foregoing specific training or certifications by project managers and employees prior to adoption
- early goal determination - this method commits to a defined end goal, product, or specified deliverable from the start of the process
- …

and some noted issues are as follows,

- lack of inherent flexibility - Waterfall method, by its underlying nature, does not permit flexibility, unlike comparable *Agile* methods
- sequential dependency - each stage can only continue as and when each of the previous stages has been completed and approved
- difficulty managing changes - if changes are required during the implementation phase, this necessitates a new design to be created and approved before implementation is completed
- …

The *Waterfall method* is a well-established project management workflow that is best suited for projects with clear, unchanging requirements and a defined end goal. However, its lack of flexibility can be a drawback in projects that require adaptability and iterative development.

structure - agile By contrast, we may consider a common alternative to the Waterfall method, *Agile*.

Agile methodology is a practice, which encourages and promotes continuous development and testing throughout a project's software development lifecycle. Unlike the previous Waterfall methodology, Agile allows for parallel development and testing. Such methodologies aim to deliver the right product through small, related, often self structured and organising, teams, which produce small pieces of functionality on a regular basis. This allows for frequent customer input and course correction as needed, another noticeable difference with the previous Waterfall method.

For *agile* based development, we might consider the following initial stages

- requirements - product owner generates the initial documentation, which includes a description of the project's needs
- design - consideration and outline of app's design, including visual UI design, where appropriate, and the app's architectural structure

Perceived benefits may be considered as follows,

- teamwork - Agile promotes effective teamwork by addressing common project challenges such as scope creep, unrealistic deadlines, and unresolved dependencies
- adaptability - common benefit of using Agile processes in software development is the ability to shift strategies quickly, without necessarily disrupting the flow of a project
- greater productivity - can lead to greater productivity and team alignment
- reduce risk - ease of adaptation, response, and a flexible approach to development may help to reduce risk

Agile development involves a number of challenges, such as the requirement for a different mindset to overall project development, cultural transformations for project teams and within a company, and the need for regular training and education in various contexts.

In effect, Agile may be perceived akin to a state of mind for project development, not a cure all prescription for general development. As such, it should feel expansive and freeing rather than confining for a project's developers and the company. The specific industry, company culture, team size, and the type of product a company is developing will, in the end, influence the appropriate chosen methodology.

Standards and guidelines

Whilst it is clearly possible to develop and code all projects and applications from scratch, without awareness, adherence, or reliance on given industry standards, it will commonly lead to significant issues and compliance errors with longterm development, updates, and ongoing maintenance.

As such, we might perceive *standards*, in the context of software development and project management and structure, as a set of guidelines or protocols, which become accepted and adopted by developers for project development.

Software development standards, for example, provide a set of guidelines and best practices for developing software. They define a common approach to software development and help ensure that software is developed in a consistent, reliable, and efficient manner

Specifically, such standards might include recommendations and guidelines for code formatting and various style guidelines. These are essential for ensuring that code is easy to read and maintain. These guidelines define how code should be structured, including indentation, spacing, and naming conventions.

Common examples include the following,

- Google's style guides - https://google.github.io/styleguide/
- JavaScript standard style - https://standardjs.com/
- GitHub's Ruby style guides - https://primer.style/
- Python Foundation's style guide (PEP 8) - https://peps.python.org/pep-0008/
- AirBNB's JavaScript style guide - https://airbnb.io/javascript/
- Angular's style guide - https://angular.io/guide/styleguide
- ...

Such style guides provide a set of standards for writing and designing code. They help maintain a consistent style, voice, and tone across a project's codebase, from a single developer to a large development team. Following recommended guidelines can improve a project's code making it easier to read, reducing cognitive load, and increasing confidence in the code's underlying authority.

However, a specific team and project, where necessary, may also customise existing guidelines to help promote consistency in their team, or simply prescribe specific guidelines to fit the project where unsupported elsewhere. In effect, they may deem it necessary to *extend* the current guidelines, akin to modification and reuse of schemas and namespaces for a specific project.

Such standards might be specific to programming languages, application development, service provision, &c. A few common examples are as follows

- ISO standards for software development - https://www.iso.org/standards.html
  - ISO 12207 for Software life cycle processes
  - ISO 29119 for Software Testing
  - ISO 27001 for Information Security
  - …
- NASA software engineering - https://swehb.nasa.gov/display/SWEHBVC/9.03+Coding+Standards
- …

Guidelines may also be applied for other aspects of software development, beyond the day to day coding tasks, including testing, bug tracking, and issue management. For example,

- NIST - Guidelines on Minimum Standards for Developer Verification of Software
- ISO standards for software development, software life cycle processes &c.
  - ISO 12207 for Software life cycle processes
  - ISO 29119 for Software Testing
  - ISO 15288 for Systems and software engineering
- …

Software standards enable interoperability between different programs created by different developers. For example, many businesses get certification in order to adhere to established standards, with ISO being the most prominent and well-known.

For each of these standards and guidelines, the underlying goal is to ensure consistency, readability, maintainability, and reliability in software development. In effect, they may serve as a *roadmap* of sorts, which is designed to help developers write and develop code in a structured and organised manner.

Resources

- Agile development: The benefits and challenges of this popular software development methodology - Scrums.com - https://www.scrums.com/blog/agile-development-the-benefits-and-challenges-of-this-popular-software-development-methodology
- Guidelines on Minimum Standards for Developer Verification of Software - NIST - https://nvlpubs.nist.gov/nistpubs/ir/2
- ISO Standards - https://www.iso.org/standards.html
- Jira Service Management - Atlassian - https://www.atlassian.com/software/jira/service-management
- NASA Software Engineering Standards - https://swehb.nasa.gov/display/SWEHBVC/9.03+Coding+Standards
- Project Manager Roles and Responsibilities for Software Projects - Project Management - https://project-management.com/project-manager-roles-responsibilities-software-projects/
- Royce, Winston. W. "Managing the Development of Large Software Systems". 1970. - https://blog.jbrains.ca/assets/articles/royce1970.pdf
- Ruby in Style - https://ruby.style/
- Standards in software development and 9 best practices - OpsLevel - https://www.opslevel.com/resources/standards-in-software-development-and-9-best-practices
- Waterfall Methodology: A Comprehensive Guide - Atlassian - https://www.atlassian.com/agile/project-management/waterfall-methodology
- What is Agile methodology? - Asana - https://asana.com/resources/agile-methodology
- What is Jira Software, and why use it? - Atlassian Community - https://community.atlassian.com/t5/Jira-Software-articles/What-is-Jira-Software-and-why-use-it/ba-p/2323812
- What is the role of a project manager in software development? - developer.com - https://www.developer.com/project-management/role-of-a-project-manager-in-software-development/