

Approaching Software Projects

Author: Mark Osborne
markosborne@pcblues.com
Date: 16.11.2024
Version: 1.3
Created: 14.6.2019

Introduction

This document describes components that make up my recommended approach to software development projects, reporting projects and other projects that require product configuration based on business requirements. The inclusion or exclusion of components in a project affect the costs and benefits to the business.

Thoughtful attention to each of the components provides compounded project-wide savings in time and money by reducing repeated design, development, testing and documentation tasks as the life cycle of the project progresses.

Some of the components are not necessary to complete to MVP stage (Minimal Viable Product) for some types of projects, so long as the potential risks are accepted by the business. All the components are recommended to complete medium-sized projects and larger (i.e., 100 hours+)

The components are roughly in the order that they will contribute best to the project. Some will be works in progress throughout the project. It is valuable to start them all after the creation of the Project Repository so they can be expanded and modified as design and project decisions are made and re-made. It encourages the developer to save important information - and realisations as they occur - in suitable documents for future use.

The components form an important reference for the project - they answer questions that are necessary for a good project result. Not answering questions at the appropriate time afflicts a project with greater instability, more bugs, failures, downtime and business risks, and more expensive fixes and extensions well into its production lifetime.

Component	Explanation	Answers Questions	Benefits With This	Costs Without This
Scope	<p>Provided by the business, it contains:</p> <ul style="list-style-type: none"> • identification of the need for the project • business goal of the project • definition of the project's success • relevant personnel (e.g., stakeholders, subject matter experts etc.) 	<p>Why does the business need the project?</p> <p>Who is running the project?</p> <p>Who understands the project?</p>	<p>Focuses the project and any related decisions in the right direction.</p>	<p>Any future decision in the project may work against it.</p>
Project Development Notes	<p>Maintained by the developer, it includes:</p> <ul style="list-style-type: none"> • ideas • locations of code, documentation, and resources • design decisions accepted and rejected • investigations successful and failed • assumptions and shortcuts taken to save time (e.g., To Do Later lists) 	<p>What has the developer done?</p> <p>What design issues has the developer confronted?</p> <p>What shortcuts and assumptions has the developer made?</p> <p>What has the developer deferred?</p>	<p>Time-saving development reference for any developer working on the product.</p>	<p>Reliance on memory for previous work, discussions and decisions made on the project.</p>
Development Checklist	<p>This document, created by the business project team, contains important information and checkpoints required for the project.</p>	<p>How can another developer pick up the work?</p>	<p>Another developer can answer the initial set of questions required to review or continue work on the project.</p>	<p>The initial developer needs to be available to answer questions for another developer to review or continue work on the project.</p>

Component	Explanation	Answers Questions	Benefits With This	Costs Without This
Project Repository	<p>A single documented place for documentation and another for the source code.</p> <p>It will contain:</p> <ul style="list-style-type: none"> • all project documentation and related communications • related 3rd Party guides and documentation • business-provided configuration, data, etc. 	<p>Where can I find information relating to the project, its requirements, and decisions?</p>	<p>Single place to find answers to project-related questions.</p>	<p>No single source of truth regarding decisions and directions of the project.</p>
Time Records	<p>A record of time spent by the developer on each component of the project, broken down into tasks where necessary for greater detail.</p> <p>Can be recorded against the Project and/or Development plans.</p>	<p>What has the developer completed?</p> <p>How long does development have to go?</p> <p>When will a stage of the project be complete?</p> <p>When can a stage of the project begin?</p> <p>How much time has been spent on the project so far?</p>	<p>Ability to predict development completion time and cost from the current state of the project.</p> <p>Ability to determine if the development costs are going to outweigh the benefits of the project.</p> <p>Ability to determine if important changes of direction need to occur.</p> <p>Ability to make informed adjustments to the project plan.</p>	<p>No knowledge of the current development state of the project.</p> <p>No ability to project the total cost of development for the project.</p> <p>No knowledge of realistic development completion times.</p>

Component	Explanation	Answers Questions	Benefits With This	Costs Without This
Requirements	<p>Maintained by the business, contains details of the required inputs, outputs and outcomes of the project.</p> <p>Should include documentation of how the user interface is expected to display and behave.</p>	<p>How is the software meant to operate?</p> <p>Will the project meet the scope?</p> <p>Is anything missing from the requirements?</p>	<p>Developer has clear instruction for what the software should provide.</p> <p>Business has a clear description of what to expect from the developer.</p>	<p>Developed functionality may not meet the business requirements or address the scoped needs.</p>
Project Plan	<p>Maintained by the Project Manager, or developer in their absence.</p> <p>Breakdown of tasks and timing required to deliver all aspects of the project to completion signoff.</p>	<p>How is the project progressing?</p>	<p>Ability to predict project completion time and cost from the current state of the project.</p> <p>Ability to determine if important changes of direction need to occur.</p> <p>Ability to make informed adjustments to the project lan.</p>	<p>No knowledge of the current state of the project.</p> <p>No ability to project the total cost of the project.</p> <p>No knowledge of realistic completion times.</p>
Development Plan	<p>Breakdown of tasks required to deliver the project from a development point of view.</p> <p>Include high-level diagrams of how the subsystems of the project will interact.</p>	<p>How will the project be developed?</p>	<p>Provides the business with some ability to gauge the size of the project.</p>	<p>Time/cost estimation is inaccurate without an initial breakdown of the required development activities.</p>
Development Specifications	<p>Documentation of data structures and database objects.</p>	<p>How can another developer understand the software?</p>	<p>Developer can accurately write software using this as a reference.</p>	<p>Unnecessary development issues and delays arise from a lack of this documentation.</p>

Component	Explanation	Answers Questions	Benefits With This	Costs Without This
Integration Specifications	Documentation of shared data and function interfaces with other systems.	How does the software interact with other systems?	Developer and vendors of other systems understand the proposed interactions between systems.	Unnecessary development issues and delays arise from a lack of this documentation.
Issue Register	To track defects, issues, and improvement requests identified during development, testing or production.	What project issues are outstanding? What project issues and resolved?	A single source of outstanding and resolved issues provides a place to consolidate similar issues, find patterns within the issues and prioritise their attention and resolution.	Duplication of reporting and investigation of outstanding issues. Inability to easily track the people involved with the issue.
POC (Proof of Concept)	Critical functionality should be quickly implemented with the tools selected to help ensure the development approach will meet the requirements. This can be thrown out when no longer needed as a reference.	Will the project work?	Avoid significant investment in the project before an insurmountable challenge is faced.	Technical challenges have the potential to derail a project or render it unable to meet the scope or requirements.
Complexity Analysis	Developer-provided analysis of the design, tools and requirements to determine the complexity of the project.	How many complex problems need to be solved?	Reducing the potential for unexpected difficulties assists the accuracy of project estimates.	The complexity of a project has a multiplier effect on the impact of costs and benefits. The business cannot analyse the risk of the project without understanding its complexity.
Project Time/Cost Estimate	Covers time and costs for: <ul style="list-style-type: none"> • hardware and software provision and licensing • developer time 	How long will the project take? What will the project cost?	Business is aware of a realistic total cost of the project.	Business is unaware of the total potential cost of a project.

Component	Explanation	Answers Questions	Benefits With This	Costs Without This
Risk Analysis	Undertaken by the business to decide if the project in its current form is worth undertaking	Is the project going to benefit the business?	Business has an understanding if the project will provide net benefit.	Business is blind to whether project will advance or hinder the business.
Project Start	Business formally signs off beginning of project with a commitment to its completion.	Is the business behind the project?	Participants in the project benefit from high-level backing of the project.	The project is assumed to be committed to without certainty.
Training Notes	Maintained by the developer for the benefit of creator of the training materials: <ul style="list-style-type: none"> • limits to features • unexpected behaviour • hidden features • known issues 	What does the creator of the training materials need to know?	The creator of the training materials will be made aware of functions and issues they may not discover during their own preparation.	Pitfalls and useful functions are missing from the training materials.
Development Test Plan	Maintained by the developer, includes tests required to: <ul style="list-style-type: none"> • guarantee functional completeness • expose design issues • test for deferred functionality/development rigour 	What areas of the software need to be tested?	Functionality and design issues that would be more expensive for the business to deal with after go-live can be predicted and tested early.	Without detailed developer testing, many bugs, functional and design issues are pushed to production. Post-production discovery of these issues causes morale issues as well as requiring more expensive mitigation.
UAT Test Plan	The User Acceptance Testing plan is prepared by the business to provide a standard for acceptance of the project's functional completeness.	What does the software need to do for the business to consider the project complete?	Business is confident that it will be able to verify that requirements and scope are met.	Business will be reliant on developer's definition of "working", "complete" and "finished "

Component	Explanation	Answers Questions	Benefits With This	Costs Without This
Training Plan	<p>Maintained by the business to ensure users of the project will be prepared to use it.</p> <p>Includes:</p> <ul style="list-style-type: none"> • required materials • users and their roles • scheduled training 	What do users need to know about the software and how to operate it?	Users and support will be equipped with everything they need to know about using the software.	Users will be uncertain how to use the software correctly. This causes anxiety, stress, and potentially costly business errors.
Training Materials	<p>Prepared by the creator of the training materials in conjunction with the developer.</p> <p>Includes:</p> <ul style="list-style-type: none"> • manuals • websites • reference cards, etc. 	What can users/helpdesk refer to, to learn about the software?	<p>Users and support know where to find information regarding how to use the software.</p> <p>The training materials become the baseline of how the software is expected to operate.</p>	<p>Users need to maintain their own notes and will use the software in different ways, resulting in potential business errors and unnecessary reporting of bugs and issues.</p> <p>Unnecessary investigation of software issues without a clear idea of how the software is supposed to operate.</p>
Deployment Documentation	<p>Includes instructions to deploy the project and make its functionality available to the appropriate users.</p> <p>Should cover development, test, production, and training environments.</p>	<p>How and where will the software/hardware be deployed?</p> <p>How will users be initially set up to access the software?</p>	Clear instruction on the location and method of deployment eases the task and ensures users will be able to successfully access the software on go-live.	<p>Technical support will be unable to maintain the software's running state.</p> <p>Users may not be able to use the software properly on go-live.</p>
UAT Test Results	<p>Results of UAT testing over the final iteration of the project to be deployed.</p> <p>Can be included in the UAT Test Plan.</p>	Is the project considered functionally complete by the business?	Business is confident that requirements are met, and the scope is achieved.	Business does not know if requirements or scope is met.

Component	Explanation	Answers Questions	Benefits With This	Costs Without This
Handover Documentation	Contains instructions for the group that is responsible for running the new system.	<p>How does the technical support team stop, start, restart, or move the system?</p> <p>How does the technical support team identify faults in the system?</p> <p>How does the technical support team maintain the user list and their permissions?</p>	Technical support can ensure the continued operation and availability of the software.	<p>When operational issues arise, technical support have no accessible reference to begin root-cause investigations.</p> <p>It will be unclear how to correctly set up and decommission users as their requirements change.</p>
Development Test Results	<p>Results of development testing over the final iteration of the project to be deployed.</p> <p>Can be included in the Development Test Plan.</p>	<p>Is the project feature complete?</p> <p>Are identified edge-cases handled correctly?</p>	The developer and business have insight to the feature-completeness of the software.	A "Happy Path" approach (one single correct operation) to declaring software functionality is "complete" or "working" is a guarantee of many expensive post- production issues.
Completion sign-off	Business acceptance that requirements have been met.	Does the business accept the project as completed?	The developer and business is clear that the project is complete.	<p>The project cannot move from a development to support phase.</p> <p>The different phases signify changes in developer availability and in responsibility for the on-going operation of the software.</p>

Component	Explanation	Answers Questions	Benefits With This	Costs Without This
Formal modification requests	Documentation of change requests, authorisation, and implementation.	<p>What has been requested to be changed/fixed?</p> <p>What has been changed/fixed?</p>	<p>A record of post-production works is available for reference.</p> <p>Business can approve of function and design changes before implementation.</p>	<p>Changes that fix one problem, but cause others are likely.</p> <p>Duplication of change requests are likely.</p> <p>Coordination between the software and associated documentation can drift.</p>