Software Development Project Management

(Detailed Overview)

0. ## Scope Estimation / Value Proposition:

Define “selling points” of project. Analyze how the work is being done now and calculate potential approximate time/money savings.
Keep development circumstances in mind as well, such as availability of a data-warehouse, APIs etc.
Also, create Timesheet using the *Software Project Management – Time Estimation* document.

1. ## Requirements Gathering:

Gather information on desired software behavior and end results. Documents to be made are:

a) Software Requirements Specifications document (SRS)
b) Preliminary Software Architecture Diagram

Only continue with the next Waterfall step after SRS document has been approved by all stakeholders (Manager, Internal End-Users). Revise SRS as often as needed.

Also, let users record their current workflow as screen-captures if applicable.

2. ## Prototyping (PoC):

Develop bare-bones, core functionality (no input checks etc.).
Present PoC to stakeholders and iterate development, adjust SRS until accurate and document current features/behavior.
Also, create interaction videos and screenshots showcasing the way of interacting with the PoC. Reiterate until approved.

3. <u>Redesign Codebase:</u>

   Redesign Codebase and Software Architecture to ensure modularity, extensibility and fast test-setup.

4. <u>Testing:</u>

   Derive tests from Waterfall step 1 and flesh out all *details*.
   Document expected results in test documentation documents.
   Test extensively and rewrite and create an Extended SRS document (ESRS) if need be.

5. <u>Integration Test:</u>

   Observe how the software integrates into the existing workflow. Software use is encouraged at this point, but not as a replacement for manual work. *Add "software is unsafe for use" notice on every UI view possible.*
   Get screen recordings of software use and check if used as intended.
   Approve this step yourself.

6. <u>Release:</u>

   Write User Manual + Troubleshooting Guide + FAQ
   Write Installation Guide
   Record video of installation and usage (crucial for fast on-boarding of new team-members)

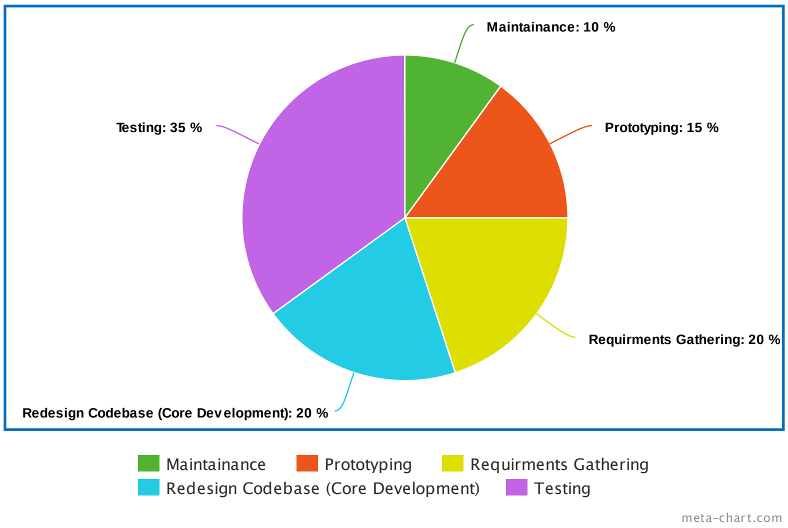
7. <u>Maintenance:</u>

   Hotfixing of bugs, up-to-date changes if things change.

8. Feature Addition:

Is the new feature in the project scope or should it be separated into another Waterfall Procedure?

Example Chart for typical distribution of time-work:


Maintainance: 10 %
Prototyping: 15 %
Testing: 35 %
Requirments Gathering: 20 %
Redesign Codebase (Core Development): 20 %
🟩 Maintainance
🟥 Prototyping
🟨 Requirments Gathering
🟦 Redesign Codebase (Core Development)
🟪 Testing
meta-chart.com