Digital Product Development Breakdowns and How to Avoid Them

More directional pivots than line-dancing night at the local country bar, “didn’t think of that” things popping up, team onboarding delays, communication stalls between on-shore and off-shore teams, finding out you can’t do **This** without **That** first, going over budget, code deployment stalls or breakdowns. Oh boy… the challenge list can go on and on.

Product development processing challenge mitigation is possible when you hire Otten Analysis Inc!

If you have been through a series of bad consultant relationships and perhaps not yet ready for a new commitment, don’t fret because I’ve still got you! Read on for great ideas on mitigating some common digital product development challenges.

Missed Requirements

Requirements can be missed when elicitation is not executed with the involvement of all key stakeholders or when requirement providers have limited exposure to new ideas or concepts (tribal knowledge) that can help ensure high-value, fully inclusive requirements.

Missing requirements can be a common issue when using domain SMEs as analysts with little to no professional analysis expertise or training. To no fault of their own, they’ll most likely only assess what they know and may not have the insight to look at other options.

To mitigate this issue, it is important to include all key stakeholders in your elicitation and to use an elicitation framework such as the 6Ms of Production, facilitated by a professional business analyst.

Mitigation Steps:

1. Collaboratively document all system components
2. Document all key aspects of each component using a SIPOC (Supplies, Inputs, Process, Outputs & Customers) and the 6 M's of Production
3. Document the high-level, future state requirements for each component
4. Highlight requirements that are not currently met or are missing (gaps to solve for)

Team Onboarding Delays

One could argue that a "delay" here is probably only an assumption and, this typically happens when the process for onboarding a new team or new team associates is not well documented for reference, agreed upon by all actors and, there are no support SLAs in place.

Mitigation Steps:

1. Document the onboarding process (stages, actor departments, work description)
2. Develop and document service-level agreements (processing time expectations) for each
3. Establish a stage-specific breakdown plan for when SLAs are missed (Who backs up who?)
4. Post and communicate the information to key stakeholders in a centralized repository (Wiki)
5. Monitor processing performance
6. Work with leadership to remove impediments
7. Apply improvements where needed

Can't Do This Without That

Missing enablers that are required to move the team forward typically happen when the product requirements are not completed well in advance (2 to 3 Sprints out), so there is not adequate time for technical leadership review. *Lacking an architectural design could play negatively into this scenario too, where technical experts are scrambling to stitch everything together later rather than sooner.*

Mitigation Steps:

1. ALLOW ENOUGH TIME FOR PLANNING AND PREPARATION (Yes, this is a thing, so why not budget for it, schedule it, and support the idea of it?)
2. Obtain a complete architectural design and specifications document for review
3. Confirm the solution scope for the upcoming release
4. Involve the solution architect, development, QA leads, project manager, product owner and/or product manager at least in the development of your Epic and Feature work items
5. Review the system components breakdown and requirements
6. Discuss what technical enablers will be required
7. Document the required dependency work (enablers) in the backlog

Communication Stalls Between On-shore and Off-shore Teams

Utilizing teams in various time zones across the globe has its challenges. One common challenge that can be easily mitigated is a stall in the development process when there is confusion on a processing step and those who can help with the answer aren't immediately available. See below for a solid plan to help omit this breakdown.

Mitigation Steps:

1. Collaboratively with the on-shore and off-shore teammates map the development and QA process the team needs to follow, step by step
2. MAKE SURE EVERYONE ON THE TEAM HAS INPUT INTO THE DETAILS OF THE PROCESS
3. Assign each step to a role on the team (Developer, QA Engineer, Dev Lead, QA Lead, etc.)
4. Set service level agreements (SLAs) for each step where applicable
5. Post the process artifact in a central location (Wiki?)
6. Review frequently with the team especially early on to add any missed steps or to apply changes as needed

Going Over Budget

Development and quality assurance teams typically don't live in the $$ space, so "budgeting" in this context applies to the development and testing time allowed for product development.

To control the budget avoiding scope creep is key. You can also add more clarity for team unification around the budget by story-pointing even the high-level work items such as Epics and Features. *For example, if an Epic has a story point value of 100 points, the budget for the features linked to the Epic must take no more than 100 development/QA days to complete.*

With this information, the solution architect, UI/UX designers & other key technical stakeholders can more easily engineer solutions that meet the budgetary constraints. At the User Story level when a developer can see they only have 3 story points to complete a story vs 8 story points their solution will probably be more straightforward and consist of fewer "shiny" things.

Mitigation Steps:

1. Each story point shall represent 1 working day
2. Story point estimates shall reflect the development and QA time required to finish the work
3. Story point Epics, Features & User Stories
4. User story points should not exceed their feature's points
5. Feature points shall not excel their epic's points
6. Make sure all on the team understand that the story point value is the budget for the work

Code Deployment Stalls or Breakdowns

We have directly witnessed breakdowns in the deployment of code when the process is not officially documented, there is a lack of actor-level service level agreements, and no clear backup strategies for process step actors that are out of the office. When this happens chaos and frustration begin to surface with stress and the feeling of failure traveling up and down stream.

Although our mitigation steps listed below won't help when your software development factory doesn't have the right infrastructure, they can help when the issues are related to a lack of an agreed-upon process, how long things should take to complete, and who needs to do it.

Digital Product Development Breakdowns and How to Avoid Them

Mitigation Steps:

1. Collaboratively with all DevOps actors map the code deployment steps the team needs to follow
2. MAKE SURE EVERYONE ON THE TEAM HAS INPUT INTO THE DETAILS OF THE PROCESS
3. Assign each step to a role on the team (Developer, Peer Reviewer, etc.)
4. Set service level agreements (SLAs) for each step where applicable
5. Post the process artifact in a central location (Wiki?)
6. Review frequently with the team especially early on to add any missed steps or to apply changes as needed

The key takeaway from this blog is your analyst(s) should be able to do more for you than create work items in a backlog. Hiring an analyst who is an expert at process engineering and modeling and has the experience to understand the importance of analyzing all system inputs and outputs, suppliers and customers can provide significant value for your product development factory.

Thank you for swinging by and enjoy your day!

Randy O'