Analysis
into the Implementation
of a Successful Project Management System
for a Hopscotch Project's Lifecycle and Software to Accompany it

for
Claire Boucher and Sofia Nolin
Founders of Hopscotch

by

Alexis Jensen
ENGL 301 Student

March 8, 2016

Table of Contents

Introduction
Hopscotch is a Mobile and Web Development Company with 20+ employees in Vancouver, BC. The company develops iOS and Android mobile apps and web applications for entrepreneurs and enterprise customers. The company is comprised of 17 developers and 3 UI UX designers as well as the 2 founders who deal mainly with the clients.

Purpose of study
Over the last five months, the Hopscotch team has grown from 5 to 20+ employees. Along with the growth of employees come the influx of projects. Before the growth, the company was handling 1 to 2 projects and communication during the lifecycle of a project was easy to manage in an organic, non-systematic way. Now with 8+ projects and 20+ employees, inefficiencies in this informal manner of management are apparent. Some inefficiencies include confusion of a project's current state and tasks being lost or not communicated.

The study gathers information on the current lifecycle of a typical project and explores the issues and software surrounding the process. The study examines staff's desired project management systems (or methodologies) and favoured project management software.

Methodology
An in-house survey was conducted to gather a snapshot of employees current methodologies during a project, explore common issues and gather recommendations on how to best implement a project information management system for the company.

The six question survey was sent to thirteen employees. All six questions were open-ended in order to avoid steering employees towards particular answers; rather, it seemed wise to cast the net wide to see how consistent the responses were.

The questions were sent using surveygizmo to ensure the anonymity of the employees. This was done to provide a safe environment where the employees could answer the questions honestly.

In addition, casual interviews were carried out to discover a more detailed look at how designers and developers work within a project's lifecycle.

Three one-on-one interviews with one UI UX designer and two developers were conducted. The goal of the interviews was to understand how the currently lifecycle of a project worked from client inception to design to implementation to QA to finished product.

Secondary sources were gathered to figure out what software development systems could successfully be incorporated into Hopscotch's more organic project workflow to add more structure. Methodologies include Agile, TDD (Test-Driven Development) and BDD (Behavior-Driven Development).

And finally web-research was conducted on what available project information management software is appropriate for software development companies of 20+ employees.

Scope of inquiry

To assess what is needed to implement the correct project information management system, the report tries to gauge the current lifecycle of of a project, determine the role of each individual in that cycle, understand how and what tools people are using to communicate with each other during a project, learn the issues people face in the current system, and find out what recommendations people have for a project overall and their project management software preferences.

Conclusion

The inquiry into the current system shows that Hopscotch staff members employ siloed methods to keep track of tasks and communicate with each other. Further there is no consistency or structure in these methods from project to project. On the other hand, there is a unified voice amongst employees for a structured project system that includes a small set of centralised tools with project management software at the center of these tools.

Analysis of the results show employees will easily adopt the correct proposed project managment system and related software, if it strikes a balance between Hopscotch's current organic process and a proven project management system.

Survey data
The survey data has been tabulated and presented in Appendix I. The total number of Hopscotch employees who answered the survey is thirteen. The founders of the company were excluded as they may be biased in their answers. Of the thirteen employees, two of them were designers and the remaining eleven were developers.

Since the questions were opened ended, the categories on the x axis of the graphs are boiled down from the survey comments. Also some employees answers contained with more than one category while others abstained to answer altogether. Therefore, the results hardly ever tally up to thirteen.

Each question on the survey has been individually analyzed below:

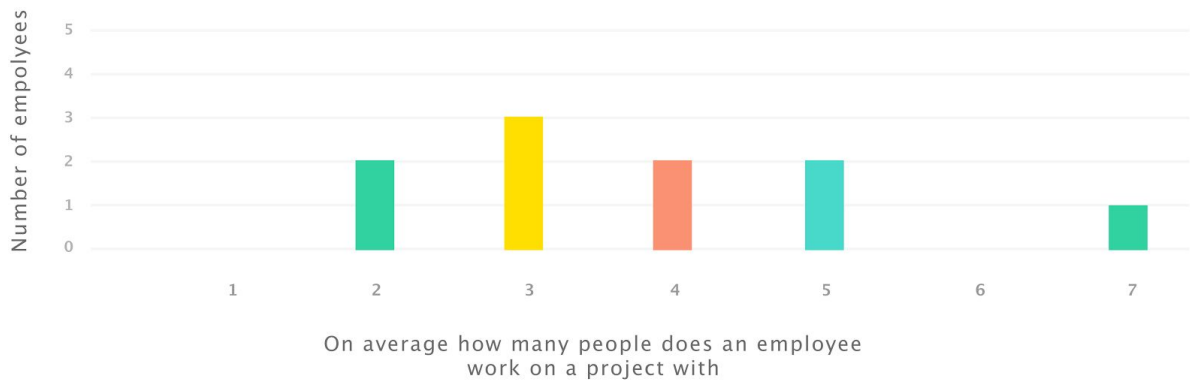
How many people do you communicate on a given project?

The average and median answers are 4.1 and 4, respectively. Anything over two is hard to manage without structure at Hopscotch's office. There are two reasons why this is the case.

The first is the logistics of the office. Many people work remotely and miss a lot of organic communication which occurs when working in one place. It is, therefore, important that communication be structured, so everyone on the project knows where to post and look for information.

The second is the nature of the work. When there is more than one developer on a project, it is important that the developers work seamlessly together. If they do not, it can lead to merge conflicts (on GitHub) and broken or buggy code.

Anything over two employees requires some sort of system in place for a project to come out successful. The findings show the average project has at least five employees. This would indicate that some sort of project management system is needed for projects to be successful.



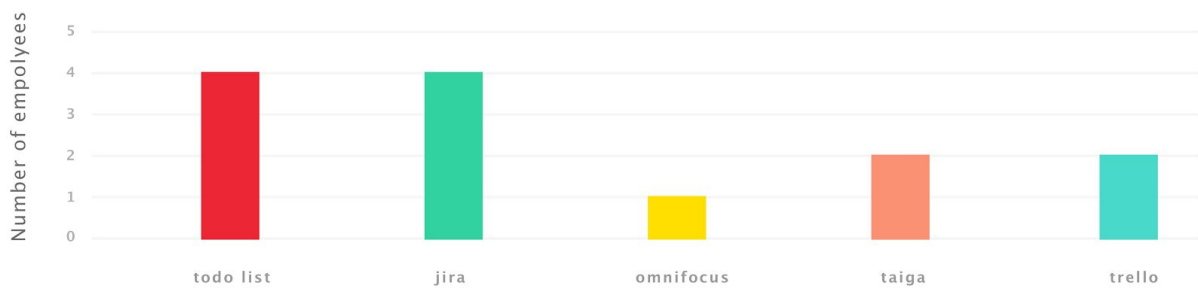
How do employees currently keep track of tasks when working on a project?

The way employees keep track of tasks of a project is evenly spread. Some use both an informal todo list system as well as structured software like Jira. One person uses two software tools: Jira and Omnifocus.

What the data shows is that all people keep track of their tasks in some form or another. Since all people are tracking tasks, it shows how necessary it is for a project be successful people need to track tasks.

A second finding in the data shows that the choices for how people track their tasks differs for almost everyone. This shows there is no clear structure in place to help declare and track their tasks.

From these findings we can conclude that in order for a project to be successful, task tracking is necessary. In addition, the lack of consistency in how tasks are tracked shows there needs to be a standard put into place company-wide.



How are people keeping track of tasks on a project

Common information management issues currently faced on projects?

The findings for what issues people face are rather unified. Many people state both information in too many places as well as lack of structure (or system in place). These issues go hand in hand, since no structure results in a lack of unity which results in information in too many places.

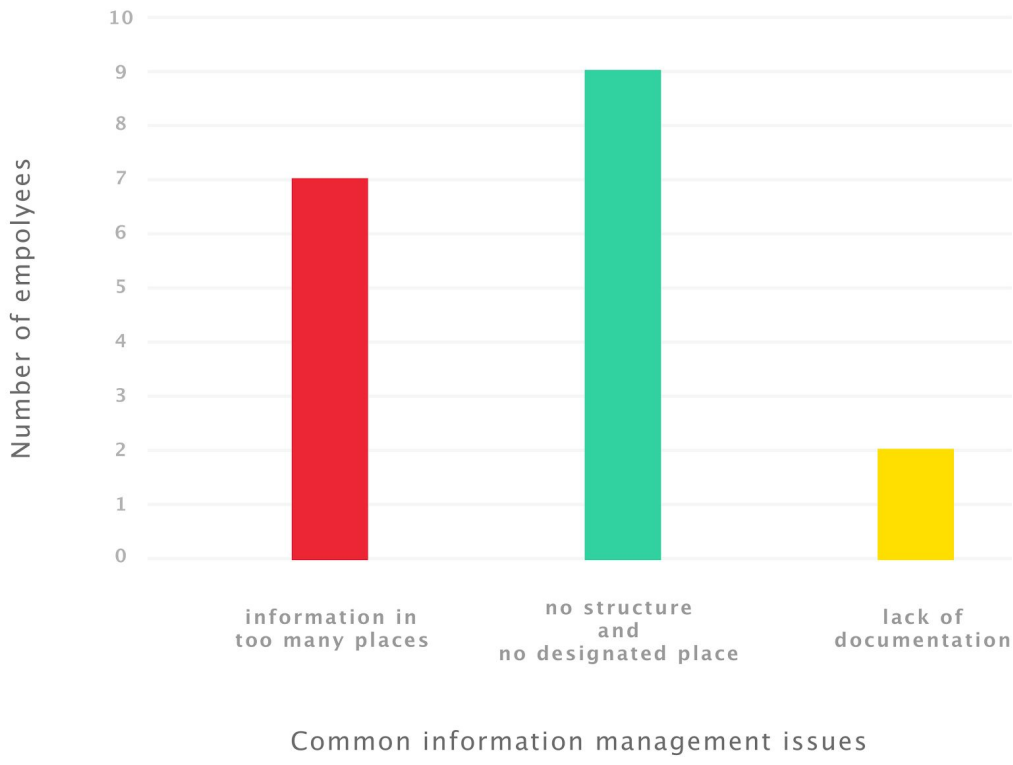
A good example of how employees describe these issues is summarized in the following comment taken from the data:

> No unity in which software to use - seems to be determined on a per-project, per-client basis. That means if we're pulled onto the project, we have to learn our way through that system/all the setup involved with that. Own standards for project management - every project has different methodologies that affect how the software is used, as well as how issues and progress are tracked. This adds another layer of complexity.

The other finding is lack of documentation which hinders employees who jump on to projects halfway through. This hindrance can cause the quality of a project's code to deteriorate as stated in this finding:

> Transitioning onto projects can be difficult due to a lack of documentation, testing, and code review, existing code is often difficult to understand. As more and more developers get thrown onto a project, the code quality tends to degrade as well.

What can be concluded is similar to what is concluded in question two but to a broader degree. Along with tasks, a standard needs to be defined for the entire lifecycle of a project - down to how a project is coded.

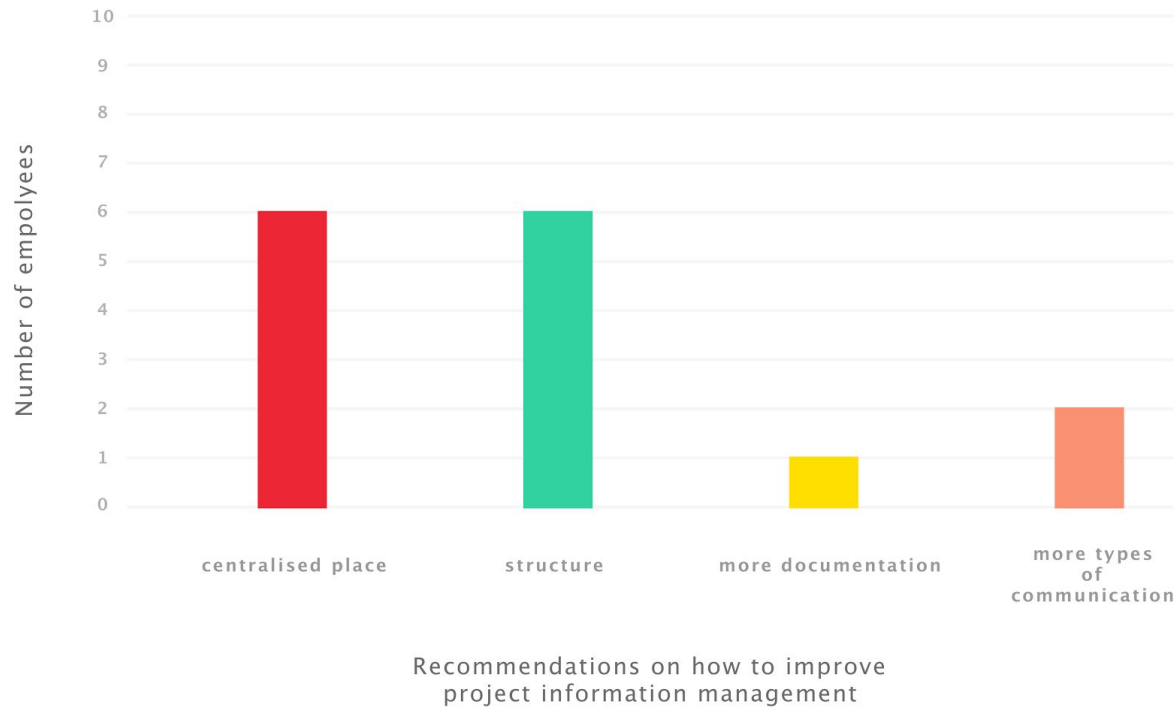


Recommendations on how to improve project information management?

Again the findings show consistent recommendations on how to improve. Almost all people suggest more structure using an Agile framework or BDD as the adopted project management process. They also recommend having the project information stored in a centralised place to communicate and manage the lifecycle of the project together.

Moreover, people would like to see more structure in the actual code. This structure includes style guides, code documentation and code review before commits.

Ironically people also desire more types of communications, mainly they wish to have more standups between people working on the same project, so they can quickly tally where people are in the project's lifecycle and help with any roadblocks.



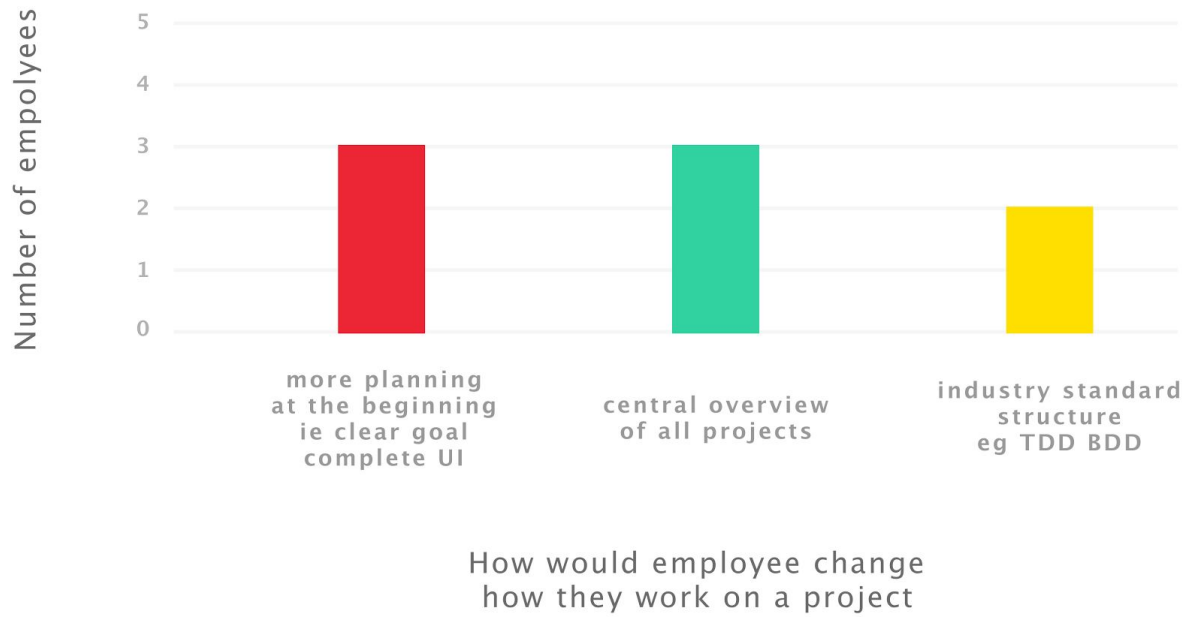
How would employee change how they work on a project

The suggestions taken from this question are a little more scattered than the above two questions. All, however, are good points to try and assimilate into the implementation of a new project management system.

The first change in behaviour would be to plan more at the beginning of a project with a clear defined goal in mind. This plays nicely into the second suggestion of introducing an industry standard like BDD. Within BDD, the participants are asked at

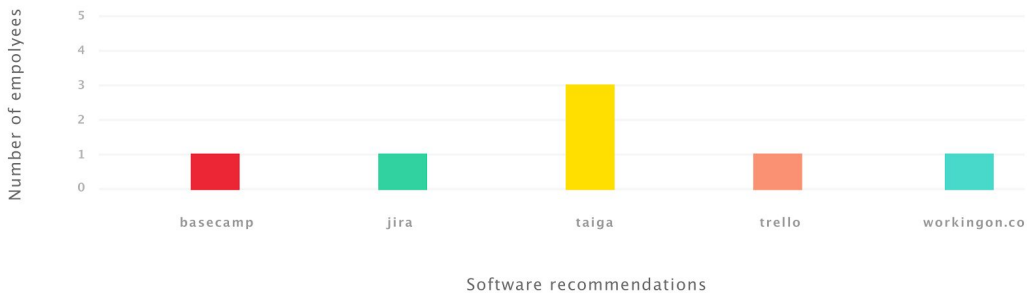
the beginning of a project to state their vision. This would be a perfect opportunity to create a system that requires a goal to be stated up front and recorded in the chosen central software at the top level of the project for all to see and plan around.

The third suggestion is having a central place where all employees can access projects. If you use something like taiga, Hopscotch can create a company account that employees can sign in to and see the list of projects: current and archived.



Software Recommendations

Some people recommend trello, jira or workingon.co but all of these recommendations had caveats attached to them, respectively: no bug tracker, too expensive, not tested enough. The most popular and the only one with no caveats was taiga as it is, “simple and flexible”.



Conclusion

Summary and Overall Interpretation of Findings

The study shows that people are no longer able to work in an organic way as was possible when it used to be two people on a project. With the growth of the company and the influx of projects, people are now working in teams of five per project. The main issues with this growth are undefined project management structures and inconsistent or siloed project management tools.

Recommendations

It is important to implement a more formal structure to a project's lifecycle at Hopscotch. This implementation is desired by staff as reflected in the anonymous survey sent out where almost all requested to have it. This sentiment suggests that implementation of a system will quickly be adopted by staff. Also recommended is the paring down of project software tools and the introduction of a centralised place, in the form of project management software. Specific recommendations are as follows:

- Choose and enforce an industry standard system blended into the current lifecycle of a typical Hopscotch project. I recommend blending Behaviour Driven Development with Hopscotch's current methodologies. The project's lifecycle would have the following steps:
    - Define vision
    - Designer cycle (before implementation)
    - Declare user stories, examples and story points
    - Define tasks for each user story
    - Code review
    - QA (Quality Assurance) cycle
- Choose and enforce style guides and documentation for project's code
- More project-based standups to define daily goals and roadblocks

- Edit down and enforce what software is used for what part of the project's lifecycle
    - Taiga for project management to store:
        - Vision statement
        - Folder for designer assets
        - User stories
        - Examples
        - Story points
        - Tasks (and who is assigned to what)
        - Bugs and their lifecycle

    - Github for actual product
    - Slack for informal communication between project members
    - Gmail for clients