

# **AGILE ONBOARDING FOR WORLDVISITZ APP**

PHASE 2 OF THE ANALYSIS

**ThoughtWorks**



# ONBOARDING AGILE – TRANSFORMING WORLDVISITZ'S SOFTWARE DEVELOPMENT PROCESS

- AGILE BENEFITS : What would Worldvisitz get by adapting the Agile Framework?
- 1. There will be continuous flow of information sharing among team members. This is very vital since the past model that we have does not have an open way of communicating.
- 2. The 4 Developers would have an ongoing channel of communication. I would also apply the 'Pair Programming' framework for the 2 US and 2 India based teams on occasional basis. This way, they could literally 'double team' on certain situations where there needs to be '2 brains and 4 eyes' that could see the code.
- 3. The main advantage of having XP is that it allows us to save costs, time, energy & eliminates unproductive activities. Developers are focused on coding. Non developers are focused on their specializations despite having constant communication.
- 4. Reduces risks related to programming & project failure.



5. Visibility & accountability is highly observed, XP has the image of enabling coders to simply write simple, straightforward & effective code. This could be improved at any time.



6. Constant feedback is a major part of our process. This means that the software is being demonstrated often and early.



7. This is a value-driven approach and thus sets fixed work times with little scope for over time.



8. There is a breakdown of project scopes into further sub-components combined with customer feedback.



9. There will be ongoing knowledge transfers among team members, especially among the 4 developers.

## **ADDITIONAL BENEFITS OF XP FOR WORLDVISITZ**

# WHY USE EXTREME PROGRAMMING OVER SCRUM & KANBAN?

- First, I'll give a summary of XP Values & how they relate to our current scenario. And how this would eventually help us achieve our target goals.
- 1. SIMPLICITY. What is needed will be done and no more.
- Given that the whole team appears to be doing a LOT of work on a very detached & individual basis, a simple information sharing would make a lot of difference, It would be brief and concise.
- 2. COMMUNICATION. Everyone is part of the team.
- The whole team must work together with a common goal in mind: to create a fully functional app within the given time frame of 18 months. There has to be minimal 'technical debt' and saving on finances, energy & efforts should be emphasized.



## WHY USE XP OVER KANBAN & SCRUM (CONTINUATION):

3. FEEDBACK. Every iteration will be taken seriously until a working product/software is available.

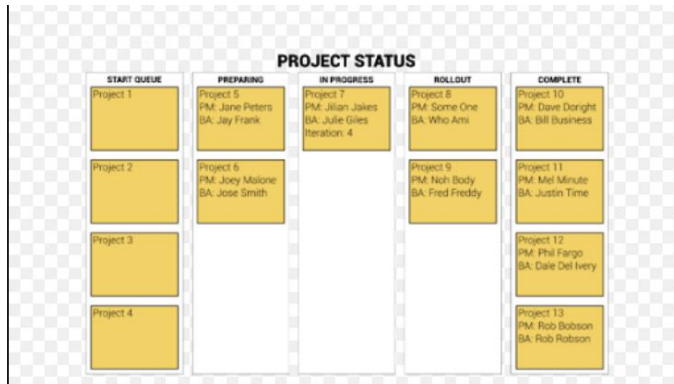
It is very vital that all 4 of our developers should have feedbacks relayed to the necessary manager. As much as possible, code and technical related issues should be relayed to the team member in charge of the coders. Such issues should also be passed on to a specific specialist if it is of concern to him/her. An example would be a Usability issue. James Crowx; our in-house UX Designer & Researcher should be informed ASAP.

4. RESPECT. Everyone gives & receives the respect deserved as a Worldvisitz Team Member.

One of the recipes for a strong & excellent product is a strong team. One of the greatest recipes for a strong team is RESPECT. Every team member would be given their chances to voice out their concerns. The concerns should be within a certain constraint that is related to the project and specific time lines and schedules should also be observed.

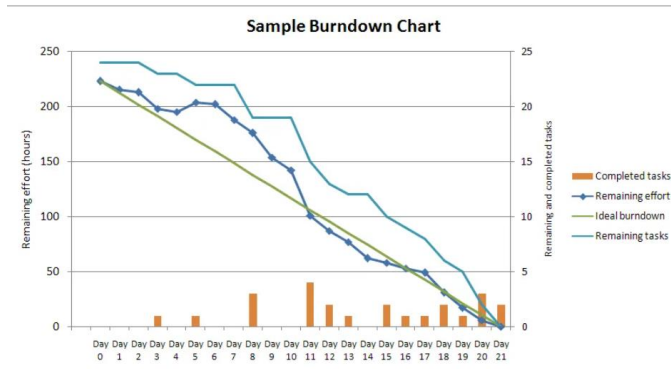
5. COURAGE. There should be transparency in stating the progress ( or lack thereof) and specific estimates.

The success of the Worldvisitz app is in having transparency regarding the current development phase. The team should learn to adopt to change whenever necessary.

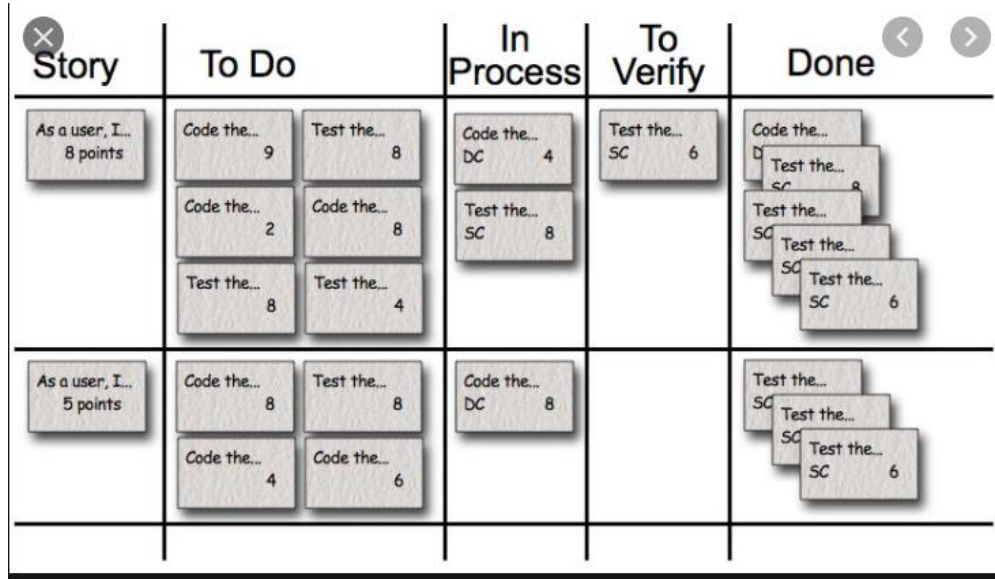


## INFORMATION RADIATORS FOR WORLDVISITZ : ENHANCING COMMUNICATION & PROPER WORKFLOW TASKBOARD & BURNDOWN CHARTS

- Information Radiators are extremely vital for our daily operations at Worldvisitz. This helps us track-down the current status not just on a 'wholistic' view, meaning as a team, but also on specialized tasks per team or individual. It helps us to be updated and look at these tasks on different granularities.
- Having a strong visual representation makes it easier and more efficient for all team members to look at their current status
- Gives all team members the numbers of iteration that are needed and on which phase should there be an improvement or upgrade
- Burndown Charts would give a linear-visual view of which tasks are completed, where the bottlenecks are & which are the main priorities
- Taskboards ensure that the relevant information is being relayed



# TASKBOARD: VISUALIZING A PROCESS BY QUICK 'STORYTELLING'



- This ensures that the proper or relevant information is being relayed to the whole team.
- Keeps the entire team focused during daily meetings, in which team members are focused on both progress & obstacles
- Very simply and flexible to utilize and allows team members to update almost automatically
- Tracks workflows, assign tasks , visualize bottlenecks and manage dependencies
- Can act as a 'focal point' where teams can meet to discuss project changes & review further updates or progress.

## TOOLS FOR DISTRIBUTED/REMOTE TEAMS

- 1. Slack
- 2. Zoom
- 3. Skype
- 4. JIRA
- The above tools are just some of the many that are available on the market, For simplicity & effectiveness, I would be suggesting Zoom and Slack.
- ZOOM: This would be utilized for tasks and projects wherein the team members involved are in distributed spaces. Although there is nothing that could beat a real time, face to face, this current COVID-19 pandemic has made us adjust to such situations. For team members which are collocated within the same towns or cities, a face to face would be done but on a limited basis and PRACTICING SAFE DISTANCING GUIDELINES ( This would be minimized as much as possible).
- SLACK: This is a tool that our team would be using regularly. I have proven this to be extremely effective with regards to communication and even sending ongoing updates.



# XP CEREMONIES : A BRIEF OVERVIEW OF XP'S 12 CEREMONIES

- 1. Planning Games : Rapidly determines the scope of the upcoming iteration. Business cases, priorities & estimations are highly considered.
- 2. Small Releases : Rapid and early deliveries are done. Quick iterations are followed. Newer versions are mostly feedback based.
- 3. Metaphors: Utilizes a simple story. This is shared by the team and minimizes too much 'technical talk/chats'.
- 4. Simple Design: All extra complexities are removed once they are being discovered. Does 'just enough design'.
- 5. Testing: 3 Steps -> 1. Develop Test Cases 2. Code Test Cases 3. Automate Test Cases
- 6. Pair Programming : 2 coders are working on 1 story where 1 'drives' and 1 'navigates'. Switching roles is highly encouraged.
- 7. Refactoring : Restructure complex & poorly structured code without affecting over-all process and behaviors
- 8. Collective Code Ownership: Anyone among the development team can work on the existing code
- 9. Continuous Integration: Tests the system. Also 'automated builds' occur in this event.
- 10. 40 -hour work week : Overtime is prohibited, Productivity and predictable results for the long term are expected.
- 11. On-site Customer/ Whole Team : At a minimum, a Product Owner and a real life user is part of the team. Enabling realistic feedback.
- 12. Coding Standards: Strict & proper adherence to code / development standards.

# XP CEREMONIES OF CHOICE: PLANNING GAMES, PAIR PROGRAMMING, REFACTORING & COLLECTIVE CODE OWNERSHIP

- 1. Planning Games is extremely useful because the entire team needs to know the upcoming iteration. The entire team would also have an overview of the actual, real world business case that Worldvisitz aims to achieve.
  - \*This specific ceremony would be done 1x to 2x weekly, preferably on a Monday before 9am (where resides). And also a 2<sup>nd</sup> ceremony on either Thursday or Friday at the same time. The 2<sup>nd</sup> time would be used to update on what the further developments or obstacles that the team encounters.
  - \*It should also be highly noted that 2 developers are based in India. So time-zone is a huge factor.
  - \* The UX Researcher and Business Analyst would both play a huge role here since they are involved with the over-all 'non-coding' aspects of both design and product architecture.
- 2. Pair Programming – This ceremony would be utilized among the 4 developers. It is highly imperative that the US team communicates with each other as much as possible. This would achieve several advantages. First, communication and information sharing is assured to occur among 2 developers in each country. This also entail for them to share their knowledge and experiences. As they say, '2 heads are better than 1'.
- \*I would also like to point out that pair programming should be done 1x to 2x weekly. Since there has been some sort of backlog due to their past practices, doing this ceremony ensures that code reviews are also done automatically. It should also be noted that pair programming may take some time, it is ideal for this project since the Worldvisitz app is a long-term project. I would highly suggest that we would rather take a bit longer in the development phase but with very minimal defects than hurry up the phase and have tons of errors.

# REFACTORING

- 3. REFACTORING- This is to continuously improve the existing code. This would remove the redundancy that may have been occurring within the entire project since there was no open form of communication among the team. This encourages the coders to have a very simple & clean code.
- \* The 2 advanced coders offshore (India) would also be serving as some sort of 'mentor' to the US coders. Since refactoring is all about continuous improvement of the code, it is highly imperative that they all have a 'near real-time' idea of how the code is being developed.
- I would be utilizing both the business analyst and UX designer here. As I have mentioned previously, these 2 are also 'designers , architects & developers' minus the coding aspect. The 4 programmers should have an ongoing open channel of communication like Slack with both BA and UX designer. All the prototypes and personas developed by this 2 would serve as a main focal point for the 4 developers.
- This ceremony would be done on a 'none-fixed' basis since the presence of code errors and redundancy could happen any time. Also given the timezone differences due to the offshore developers, plus the ongoing COVID-19 pandemic, having too much remote meetings may not always be an advantage to the entire team.

# COLLECTIVE CODE OWNERSHIP

- 4. Collective Code Ownership – the entire team would be deeply involved & responsible for the code design. This gives an opportunity for the other team members to review the outline of the product. Although the non –coders of the team may not exactly understand every line of code, they would be having their feedbacks on the UML, the diagrams and even the ‘code snippets’ that could bolster the Search Engine Optimization (SEO) of the entire application.
- This would also help in avoiding code duplication. This specific ceremony would also highly encourage the bringing in of new ideas that could be further incorporated into the ongoing iteration.
- \* This is highly suggested to be done 3x to 4x weekly and would occur via an ongoing open channel thru Slack. Ideally, there will be a ‘formal’ ceremony schedule that would occur at the beginning of the week and the remaining would be spread throughout.



## ADDITIONAL SPECIALIZED CONSULTANTS : ATTACHING THE APP TO VARIOUS BACK ENDS & ENHANCING SEARCHABILITY



- 1. SQL/ Database Developer – handle back-end and transactional processing issues
- 2. BI/ETL Consultant- Real time Predictive Analytics and ongoing data gathering from other sites/apps like Hotelsvisitz, Flightvisitz, AirlineVisitz, etc.
- 3. SEO Expert – over-all searchability and advanced digital marketing regarding site architecture





## **HIRING 3 CONTRACTUAL SPECIALISTS: SQL , BI/ETL & SEO**

- Based on my experience and training, and also looking at the over-all requirements of Worldvisitz App, I would HIGHLY SUGGEST & PRIORITIZE hiring 3 specialized contractors:
- 1. SQL Developer – This specialist would be the solely responsible for making the back end fully functional and that the over-all ‘search engine’ functionality or ‘searchability’ of the app and website becomes smooth. While I do know that our 4 developers are highly skilled, I am fully aware that they are more focused on the front end and back end design, the site architecture ( together with the UX researcher) and over-all functionality of the app.
- 2. Business Intelligence (BI) Consultant – Since Worldvisitz wants the real time predictive analytics functionality to be part of the back end, there needs to be a highly specialized individual who deals with this on a daily basis. Although the SQL developer may be skilled in this domain, I would prefer to have someone who knows a high level or predictive analytics tools. Eventually we may even have automated dashboards that updates our users regarding flights, hotel & car rental availabilities, etc.
- 3. Search Engine Optimization (SEO) Expert – this contractor would play a minimal role. Nonetheless he/she would still play a vital role. Since our business is launching a major app that would be used by a lot of end users, we have to make sure that our app and website would have very minimal ( zero if possible) regarding being searched and indexed on-line. I would highly suggest that SEO expert collaborates fully with both UX and BA.

# XP-AGILE ROLES FOR THE ENTIRE TEAM

- 1. Manager – team member that is a cross between a ‘tracker & coach’. Monitors the over-all progress of the 4 coders. This specific role would be assigned Ms. Jane Doe, our Project Manager. She would need to have a very open channel of communication not just with the 2 offshore ( which she used to do) but with the US team itself. She will be handling both the scheduling of the meeting and attend all of them. Detailed note taking is part of her tasks. All the vital notes from the meetings will then be passed to the ‘tracker’.
- \*\* The 2 consultants; SQL Developer & BI/ETL Specialist would be joining the ranks of the 4 coders, although their tasks are HIGHLY SPECIALIZED. UX Designer James Cowx would also be joining 90% of the developer related meetings and activities.
- 2. Tracker – This member is assigned to talk to each developer. Open communication is highly encouraged. This role also suggests certain solutions if he/she notices that the process is slowing down. Being an experienced BA, Jerry Holden is assigned to this role. He is very familiar with requirements elicitations, requirements gathering and other SDLC ( software development lifecycle processes). He is also familiar with SQL & Relational Databases (RDBMS) . Highly proficient in UML. Majority of Jerry’s tasks would be based on the observation notes from Jane Doe.
- 3. Coach - Sometimes nicknamed as ‘Chief Architect’ because of the nature of the job. Also tasked with mentoring less experienced team members. This role is assigned to John Smith, our in-house Product Manager. He is extremely familiar with market research & the actual product planning. It is highly imperative that he will be familiar with the entire XP process. This means that he will be constantly communicating with team members, which is the complete opposite of his former role.
- 4. Tester- testing would be done by the 4 developers, the 2 SQL/BI consultants and James Cowx would also conduct a lot of Usability testing. But this task would be mainly headed by our tester; Kathy Qualls.
- 5. Customer – ideally should be filled in by an actual end user. We can consider using a few actual stakeholders . Most of the time, this role would be filled by our UX researcher James and also by Holly Vogt. Since Holly and James have been having some form of information-sharing. This role would be very ideal for them.

# APPENDIX: CONFIDENTIAL PERSONAL NOTES FOR MAXIMUM TEAM PERFORMANCE

## ■ INITIAL PHASE: ON-LINE LEARNING FUNDAMENTALS OF KEY ASPECTS & SKILL SETS FOR MAXIMUM RESULTS

- As an Agile Consultant, I have quickly observed that the entire team has very minimal knowledge of Agile Principles. I have interviewed the 4 developers. They were fully aware of Agile and its sub umbrella methods. But they never suggested to use it because they all had the impression 2 impressions: either :
  - 1. the project would go extremely well due to the 'expertise of both the Project Manager and the Product Manager' or
  - 2. it would eventually be a total disaster and they might as well just collect their pay checks.

Based on my initial interviews which were based both on individual, group and sub-groups, I am highly suggesting that the entire team would undergo a quick on-line course on Agile Foundations. **\*That Agile course from Udacity is highly recommended.**

**\*\* The skills gap that I observed were:**

1. Over-all knowledge of Agile Principles
2. XP Focused lifecycle
3. Fundamentals knowledge of User Research ( with the exception of our UX Designer & BA)
4. Fundamental knowledge of the entire Software Development Life Cycle (SDLC) -> exceptions were the 4 coders, BA , UX designer
5. Fundamentals Knowledge of Relational Databases / Database Development Life Cycle - > exceptions were the 4 coders, BA , UX designer

**\*\* I am also requiring the entire team to undergo basic training on the fundamentals of UX, with a strong focus on Usability. The non coders are encouraged to learn the very fundamentals of Software Development Lifecycle (SDLC) and relational database design (RDBMS). I am extremely confident that this would further enhance the over-all functionality of this team.**

**\*\* The 3 external consultants (SQL , ETL & SEO) are not strictly required to attend all the ceremonies and training. They would be mostly communicating with me and Holly Vogt , the SME. The reason for this is that their skill sets are highly specialized only to certain parts of the entire process.**

**\*\* The MAIN ANTI-PATTERNS observed were the super rigid structure that they formally have. Other than this, it is the lack of knowledge transfer among all team members and the lack of proper coding standards for the 4 developers. The former practices involves having communication channels only thru selected members . This is being corrected by utilizing the new framework and also by having a flexible coaching methodology.**



## UTILIZING THE DREYFUS MODEL: IDENTIFYING SKILLS GAPS & WEAKNESSES

- Based on several assessments which were composed of both group meetings and several ‘quick sprints’ or 1 on 1 interviews, the entire team’s over-all skill sets would range from advanced beginners to proficient levels. Everyone in the team has several years of experience in their own expert domains.
- The training would initially be divided into both several modes of delivery: face to face group mentoring ( for a few who are within my distance), remote mentoring via zoom and slack and also several on- demand courses in order to keep the team updated. This way, every team member would have various ways of improving their knowledge of the Agile processes with a very strong focus on XP Model.
- Some antipatterns being observed are : extremely rigid form of communication, Jane Doe decides the over-all planning & task assigning, the early product planning of Jon Smith , the very late user research feedback received by Uxer James Cowx , the 4 developers working solely without knowledge transfers and last but not the least, James Holden (BA) freezing the requirements early on without getting much feedback.

LEVEL	DESCRIPTION
NOVICE	Needs to be told what to do specifically do. No context nor idea of any given task.
ADVANCED BEGINNER	Has some fundamental knowledge, needs more guidance
COMPETENT	Asks questions about projects/tasks, can see the consequences or certain actions
PROFICIENT	Knows to validate & give priority. Reliant on rules
EXPERT	Has high level of over-all domain knowledge & skill sets. Asks very minimal guidance; only on extreme situations



# BONUS: THE ARC OF COACHING BRIEFLY EXPLAINED

- *I would briefly explain the 'Arc of Coaching' as I deem it suitable for our current situation: This is a simple & fundamental plan that provides a very open channel of communication. This was conceptualized to ensure that conversations within the team are highly focused and that productivity is observed. . . In a relaxed manner. Below are 4 steps to approach*
- 1. Why are we having such conversation(s): What are the key metrics or factors why this is occurring? What are the key priorities & driving forces behind this conversation. Is there an urgent need for an upgrade? An update?
- 2. What are our shared outcomes? : What do we want to create out of this 'meeting'? This is better to be narrowed down & more specific.
- 3. How would / should the conversation look like? Would this conversation be more of a strategic or tactical approach? Would it be a light & friendly conversation to celebrate a 'recent victory'?
- 4. Exiting the conversation: How do I end the conversation/meeting? What are the possible next steps for upcoming discussion?
- **\*\* I would also like to add a 6 step methodology to maximize the 'Arc of Coaching':**
- 1. CHECK -IN : An informational chat that enables both the coach & coachee to establish trust. This should normally be limited to a maximum of 5 mins.
- 2. CONVERSATION PLAN: After the initial check in, there should be a plan on how the conversation would turn out. Would there be a specific structure? Main topic?
- 3. PREVIOUS COMMITMENTS DISCUSSED: Follow up questions should be highly considered. Asking the coachee " How did that last session go?"
- 4. COACHING STANCES & APPROACHES MUST BE OBSERVED: Active Listening & Questioning is vital. What are the alternative plans should Plan A or B didn't work.
- 5. DETERMINE THE NEXT STEPS: Aiding the coachee on what steps should be done next. What additional methods should he/she employ? Back up plans?
- 6. REFLECTION :Reflect and make a quick review or summary of the entire session. This would give both coach & coachee a clearer view on the target goals.

## BONUS: WHAT IS SCRUMBAN?

- SCRUMBAN is a project management methodology or framework that combines specific features of 2 popular Agile methods: Kanban & Scrum. The structure and core functionalities of these 2 methods are merged to make teams 'more flexible & fast' aka 'AGILE' , efficient & more productive.
- SCRUMBAN merges these key components: the predictability and structured framework of Scrum PLUS Kanban's openly flexible and continuous processes. This method gives teams within organizations .
- How does Scrumban Function? Application of Kanban's workflow visualization & flexibility PLUS Scrum's flexible aspects.
- 1. Creating a Scrumban Board – to be used as a primary workflow tool. Each discrete phase of the process has to be exactly marked
- 2. Set work-in-progress limits – always establish a LIMIT on how much work can a team do at a specific timeframe. Goals must be very realistic.
- 3. Team Priorities should be based on the boards – the team decides which team member does a specific task. Certain tasks are assigned to members due to their specialized skill sets.
- 4. 'Planning Poker / Estimation of Story Points' - each team can only work on a pre-defined set of projects & tasks. A Scrum team estimates how long each development would take. But with Scrumban, the work is continuous & NO TIME-LIMIT is observed. Prioritizing the most important projects is the main focus.
- 5. Setting up of daily meetings – may include daily stand ups , a series of short meetings to encourage team bonding and having constant updates on what the vital aspects of the project. Also gives team members a 'visual sense' and 'personal touch' on how each team member is currently performing regarding project tasks.