

WORLD VISITZ — SELLING THE LEADERSHIP TEAM

PART 1 (AGILE FOUNDATIONS): HOW WORLD VISITZ CAN BENEFIT FROM AGILE SOFTWARE DEVELOPMENT

PART 2 (AGILE ONBOARDING): THE OPTIMAL AGILE FRAMEWORK FOR WOLRDVISITZ

ThoughtWorks®



HOW AGILE BENEFITS AN ORGANIZATION

- AGILE is an extremely flexible & powerful tool for software development. It does provide process & efficiency benefits to the entire development team and it also provides business-case benefits to the organization as well. The most common pitfalls of any product development team are being addressed & dealt with in a very flexible and systematic way. The following are 8 key benefits of the Agile Framework:

1. IMPROVED QUALITY	Breaks down projects into manageable units, enables a team to focus on high quality development, testing & collaboration
2. FOCUS ON BUSINESS VALUE	An increased focus on business-strategic values. Involves all/most of stakeholders in the development process. Enables the team to understand the most vital aspects & focuses on the most important issues/ topics
3. FOCUS ON USERS	Utilizes user-stories with business-focused acceptance criteria. Provides an opportunity to beta-test the software
4. STAKEHOLDER ENGAGEMENT	Involves different types of stakeholders during different iterations and different steps of the project. Promotes high degree of collaboration. Provides more opportunities for the team to grasp the business values.
5. TRANSPARENCY	A unique opportunity for both clients & customers to be involved throughout the project. Includes prioritizing features, iteration planning & review sessions.
6. EARLY & PREDICTABLE DELIVERY	An opportunity to release or beta test a software earlier than planned, if there is sufficient business value.
7. PREDICTABLE COSTS & SCHEDULES	Improves decision-making about the priority of features and the need for additional iterations.
8. ALLOWS FOR GROWTH & CHANGE	Provides an opportunity to constantly refine & reprioritize the over-all product backlog. New backlog items can be planned for the next iterations.



DIFFERENTIATING AGILE FROM WATERFALL METHOD

	AGILE	WATERFALL
1. PROCESS APPROACH	Very Flexible	Structured, Very Rigid
2. ARCHITECTURAL PROCESS	Very Flexible	Structured, Very Rigid
3. TESTING PHASES	Allows in updates in all phases	Only after the build phase
4. DEVELOPMENT CYCLES	Multiple Sprints/ Different Iterations	Only one cycle
5. NUMBER OF PHASES	Multiple (depending on needs)	Only one
6. STRUCTURE OF DEVELOPMENT	Extremely Flexible	Very strict adherence to step-by-step process
7. SITUATION IDEAL TO USE	Processes involving constant changes	Projects that have VERY CLEARLY defined set of tasks

PRESENTING THE AGILE MANIFESTO : 4 VALUES & 12 PRINCIPLES

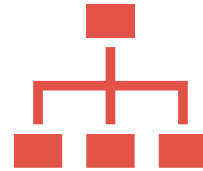
***"IT IS BETTER TO BUILD A SUCCESSFUL PRODUCT THAN
TO BUILD A PRODUCT SUCCESSFULLY"***



The Agile Manifesto has 4 main values that it lives by. Practitioners of this framework are highly encouraged to utilize these methods in real-life business cases.

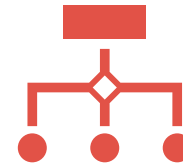
1. 'Individuals and interactions over process and tools.'

- It highly suggest that we should focus more on our people, the entire staff that is responsible for creating and eventually launching the 'Worldvisitz App' soon. This step states that our focus should be on competent personnel and that ongoing communication is key. Processes and tools are only given importance if they are helping the team's performance.



2. 'Working Software Over Comprehensive Documents.'

Traditional methods of project / product management gave a lot of emphasis on a lot of documentation. Some of these are features, specification, layouts, requirements, test cases, etc. This results in an outdated documentation which may not be even needed. This means a lot of wasted time , energy and other resources.



The Agile Manifesto highly favors feedbacks from working or functioning software.

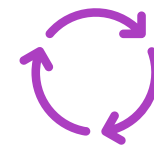
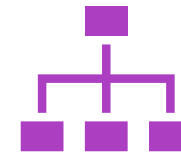
All the basic features should be created first, then be given to the end users & stakeholders. The feedback received from the users would then be used for further development & improvement of the Worldvisitz' app prototype.



3. 'CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION.'

The traditional method was very linear : Contract negotiation > Further Changes > Project Completion.

This simply meant that user and customer feedback was usually neglected during the entire SDLC (Software Development Life Cycle). Agile prefers to have a constant feedback all throughout the product development cycle. This ensures that any changes or upgrades made were relevant. Whatever is being developed is also being seen by the customer/ stakeholders/ end users.



4. 'RESPONDING TO CHANGE OVER FOLLOWING A PLAN.'

*** The shortness of iteration means priorities can be shifted & adjusted regularly. Features could be added constantly into the next iteration.**

THE AGILE MANIFESTO HAS 12 MAIN PRINCIPLES THAT IT LIVES BY. PRACTITIONERS OF THIS FRAMEWORK ARE HIGHLY ENCOURAGED TO UTILIZE THESE METHODS IN REAL-LIFE BUSINESS CASES.

1. 'Our Highest Priority is to Satisfy the Customer Through Early & Continuous Delivery of Valuable Software.'

- At the end of the day (lifecycle), what matters is a properly working Worldvisitz Mobile App that is fully functional and integrates with the necessary back end. This is why there is heavy emphasis on early feedbacks and corrections.

2. 'Welcome Changing Requirements, Even Late in Development. Agile Processes Harness Change for the Customer's Competitive.'

- End users learn more about the end product by having constantly seen the potential app. They also get a 'feel' of the end result.

3. 'Deliver Working Software Frequently, From A Couple of Weeks to A Couple of Months. With A Preference to the Shorter Timescale.'

- Frequent delivery of product is highly encouraged. The more frequent the delivery, it becomes more Agile. Risks are mitigated. We also get faster feedbacks from our stakeholders/customers. This means that changes to the product could be brought in before it's too late.

CONTINUATION : THE 12 PRINCIPLES OF AGILE (#’S 4 TO 6)

4. *‘Business People & Developers Must Work Together Daily Throughout the Project .’*

- Business and technical aspect are at the same page. This also gives the developers & solutions architects (if applicable) an ‘end users’ view of the product/ prototype.

5. *‘Build Projects Around Motivated Individuals. Give Them the Environment & Support They Need, and Trust Them to Get the Job Done.’*

- Motivated staff are highly more likely to get the job done and with BETTER results. A good environment yields superior products.

6. *‘The Most Efficient & Effective Method of Conveying Information to & Within A Dev’t Team is Face-to-Face Communication .’*

- Written communications and memos are usually filled with ambiguity. It sometimes causes confusion too. Face to face (if non remote nor non distributed) makes product development faster and more precise. If face to face is not possible, as what is happening with our current COVID-19 globally, then tools like Slack, Zoom/Skype/Google Hangouts and JIRA would make this possible.

CONTINUATION: 12 PRINCIPLES OF AGILE (#S 7 TO 9)

7. *'Working Software is the Primary Measure of Progress.'*

- Measurement of the product's over-all improvement should be based on the actual working software. Agile emphasis that:
- A software is not finished when it is successfully tested & delivered, it is finished when it is tested & accepted by the end user.

8. *'Agile Processes Promote Sustainable Development . The Sponsors, Developers & Users Should Be Able to Maintain a Constant Pace Indefinitely.'*

- Sustainable development should have a decent pace, just like 'running a marathon'. Agile emphasizes that this is NOT a sprint. A team should not exhaust their efforts and energies right away. Our team at Worldvisitz should be open and be able to endure the process on a longer timeframe.

9. *'Continuous Attention to Technical Excellence & Good Design Enhances Agility.'*

- Our team should be very observant of proper design principles and practices. This is what brings 'craftsmanship' into fruition.

CONTINUATION: 12 PRINCIPLES OF AGILE: #S 10 TO 12

10. *'SIMPLICITY -- > The Art of Maximizing the Amount of Work Not Done – Is Essential.'*

- It is highly imperative to keep the process simple. Overly complex code and frameworks would eventually cause confusion. It is much better to have a simple product/ app that lives up to the expectations and serves the customers needs. Contrast this with a product/app that tries to do 'everything' but fails in most parts; if not all.

11. *'The Best Architectures, Requirements & Designs Emerge from Self-Organizing Teams.'*

- This states that our team should manage associated responsibilities & timelines. Self organizing teams are highly recommended to have a 'mentor' (this person could be 1 of the existing team members). Collaboration, teamwork, competency, motivation, regular growth and commitment are highly encouraged.

12. *'At Regular Intervals, the Team Reflects on How To Become More Effective, Then Tunes & Adjusts Its Behavior Accordingly.'*

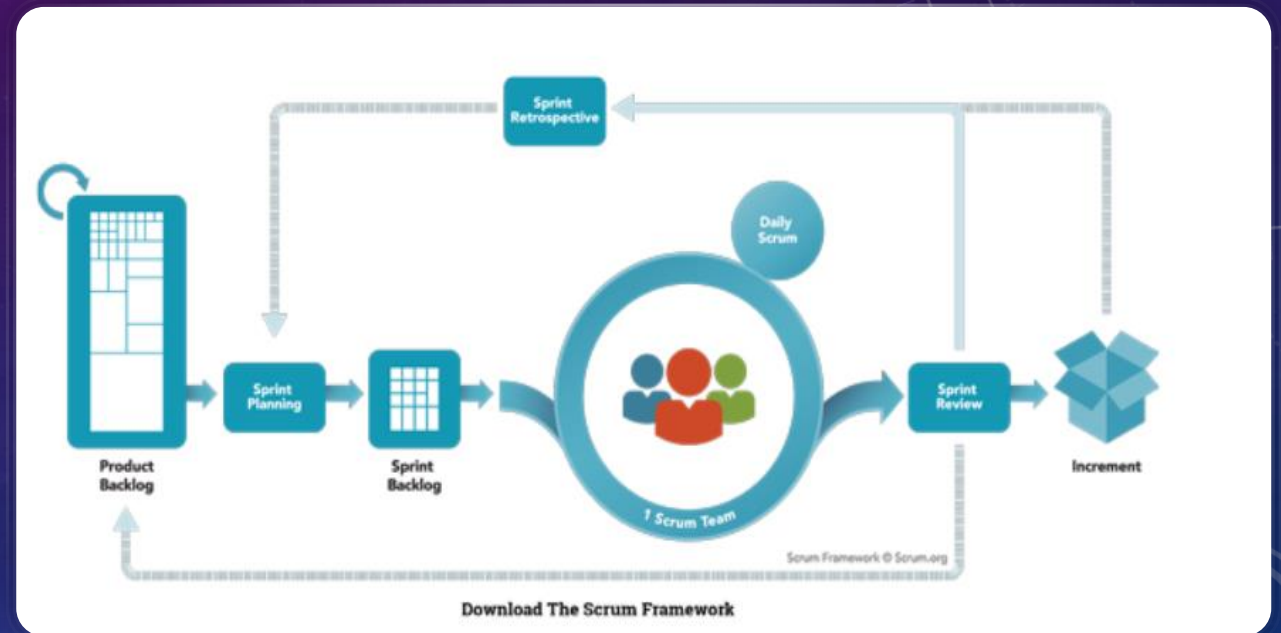
- Continuous Improvement and productivity is highly encouraged. Agile believes that when a team behaves accordingly and is observant to changes and possible improvements, then the team becomes more effective. Productivity is relatively increased.

PART 2: THE OPTIMAL AGILE FRAMEWORK FOR WORLDVITZ (SCRUM, KANBAN & XP)

A BRIEF OVERVIEW OF THE 3 MAIN AGILE FRAMEWORKS:

1. SCRUM

- A lightweight, simple to understand and difficult to master framework. Utilized to address complex adaptive problems.
- Highly emphasizes simplicity & implements the scientific method of 'empiricism'. Focuses on people rather than tools and documentations.
- Scrum Teams are assigned to the following members: Product Owner, Scrum Master & Development Team Member



SCRUM EVENTS & ARTIFACTS

- **SCRUM EVENTS:**

1. **Sprint** – No changes are made during the Sprint cycle that may appear to ‘harm’ or endanger the Sprint Goal
2. **Sprint Planning** – time boxed to a maximum of 8 hours for 1 month sprint.
3. **Daily Scrum** – Focuses on ‘Inspection’ and ‘Adaptation’. Aims to improve communications , eliminate unnecessary meetings and identify obstacles & potential hurdles
4. **Sprint Review** - done at the end of a Sprint in order to further inspect if the Product Backlog needs some modifications. The Product Owner explains what the backlog is, what is ‘Done’ and what is ‘Not Done’. The Development Team discusses what went well and what needs improvement.
5. **Sprint Retrospective** – conducted after the Sprint Review & prior to the next Sprint Planning. Team discusses: what went well, what can be improved, what steps are needed in order to improve for the next phase.



Product Backlog - an ordered list of everything that is known to be needed in the product. Considered as the single source of requirements should there be further improvements and/or upgrades. It is the Product Owner who is responsible for this artifact including availability, ordering and contents.



Sprint Backlog – It makes all the necessary & vital work that the Development team deems crucial. This normally includes 1 high priority process improvement that is usually identified in the previous 'Retrospective' meeting.



Increment – this is defined as the 'Sum of All the Product Backlog' items that were done during the initial Sprint. It also includes the value of the increments of all previous Sprints. This step is usually defined as a towards a goal or vision. The main purpose of each specific sprint is to yield potential functionalities

SCRUM ARTIFACTS

KANBAN – WORKFLOW FOR MANAGING, IMPROVING & DEFINING SERVICES THAT DELIVER KNOWLEDGE WORK



Initially this method started out as a Lean Manufacturing Framework , which was originally conceptualized and used by Toyota Production System. Originally used only for manufacturing operations, it was later 'adapted and adopted' for other development processes. Most notable of all was the software development industry.



Kanban has 4 Core Practices:



Principle 1: Start with What You Do Now



The flexibility allows it to be applied on top of other workflows , systems and business processes. This while not disrupting any patterns nor ongoing workflows.



Principle 2: Agree to Pursue Incremental, Evolutionary Change

Having massive nor sweeping changes is absolutely not a part of Kanban. The contrary is being promoted, which means having small, incremental changes and gives the organization enough time to adapt/adjust.

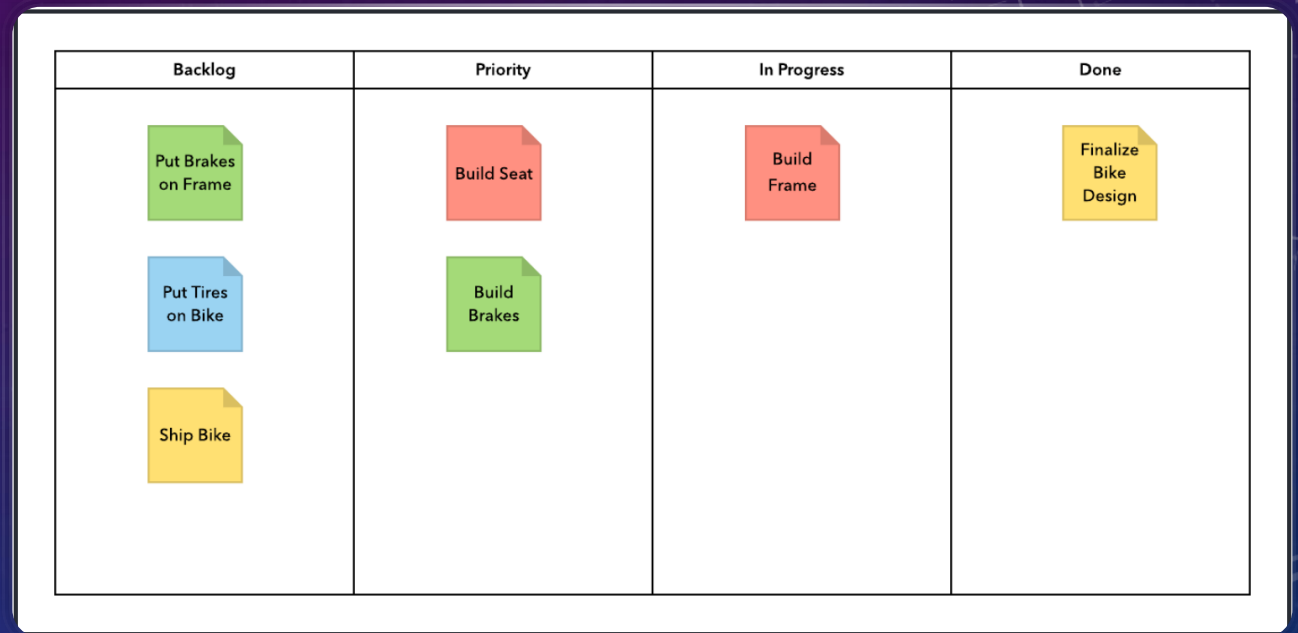
KANBAN PRINCIPLES 3 AND 4

Principle 3: Respect the Current Process , Roles & Responsibilities

This framework duly recognizes that most organizations already have their own internal frameworks. Everyone also has an ongoing structure which consists of roles, specific responsibilities and job titles.

Principle 4: Encourage Acts of Leadership at All Levels

This was the latest principle being added to Kanban. This part encourages having the 'power' to be distributed across the entire team. The Japanese term 'KAIZEN' or continuous improvement (CI) was highly encouraged.



EXTREME PROGRAMMING (XP)

Proven to be very successful in a lot of corporations and organizations.

It stresses on customer satisfaction and CX/UX. It wants to 'deliver software as you need it.'

Improves a software lifecycle project in 5 ways: communication, simplicity, feedback, respect & courage.

XP practitioners encourages constant communication with customers/stakeholders and fellow programmers.

Keep design as SIMPLE and CLEAN as it could be.

Take pride in its 'simple rules' and direct to the point approach on the software development lifecycle.

Can also be used to expand ongoing Scrum or other Agile methodologies.

The 'rules of XP' revolves around : Planning, Managing, Designing, Coding & Testing.

The 5 XP Principles are: Rapid Feedback, Assumed Simplicity, Incremental Changes, Embracing Change, Quality Work

EXTREME PROGRAMMING (XP) – MY RECOMMENDATION

- Upon doing a very thorough review of WorldVisitz' current situation, as an Agile Consultant & Specialist, I have decided that hands down, the most viable framework is Agile XP (Extreme Programming).
- I'd like to quote the 'founder' of the XP methodology, Mr. Kent Beck:
- 'XP is a lightweight methodology for **small to medium sized teams** that are developing software in the face of **vague or rapidly changing environments.**'
- Extreme Programming or XP lives by its 5 Main Core Values:
 - a. SIMPLICITY
 - b. COMMUNICATION
 - c. FEEDBACK
 - d. COURAGE
 - e. RESPECT



A DEEPER LOOK INTO XP FRAMEWORK



A. *Simplicity*

- *Reduces the complexities that are non-vital to Worldvisitz app*
- *Minimizing or eradicating features which are non critical*
- *Aims to maximize the value for the project*

B. *Communication*

- *Stand Up meetings are highly pivotal and flexible*
- *Ongoing Transparency in communications and feedback*

C. *Feedback*

- *'Failing Fast' is encouraged and then further move on for improvement*
- *Every bit of iteration is taken seriously*
- *Every working result is shown*

CONTINUATION: A DEEPER LOOK INTO XP

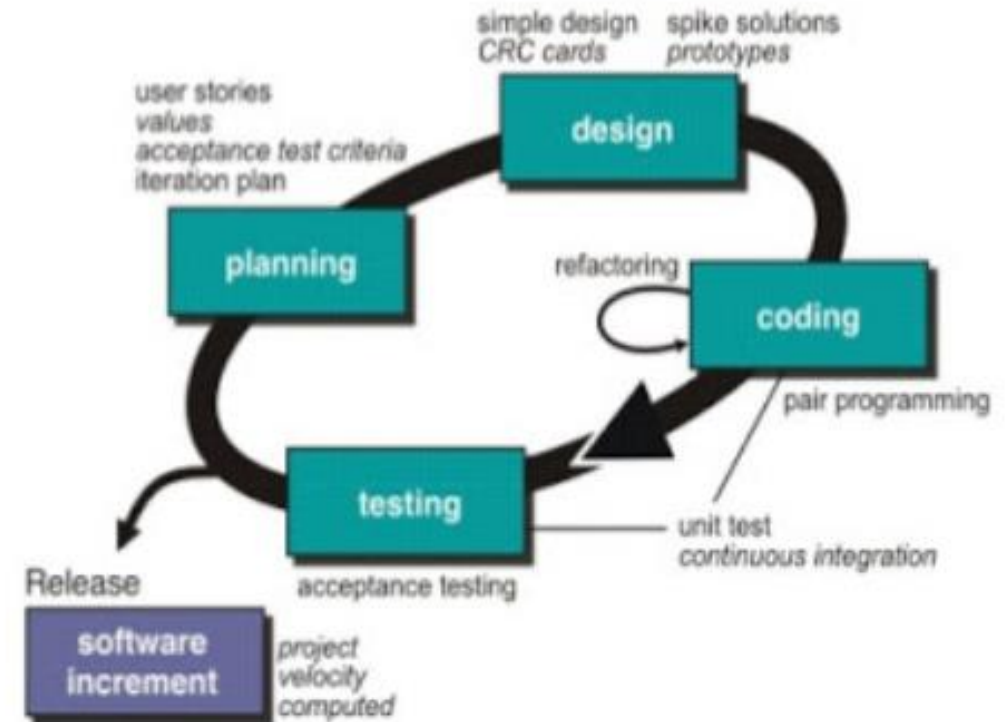
D. Courage

- *Allows a team to bravely tell the truth, the actual situation is prioritized*
- *Code sharing and communication is highly vital by means of 'PAIR PROGRAMMING'*

E. Respect

- *Highly encourages professionalism and open communication among team members*
- *Every individual within the team is held accountable*
- *High Importance is given to PAIR PROGRAMMING*

Extreme Programming (XP)



HOW AGILE (XP) SOLVES WORLDVISITZ' ISSUES & BUSINESS CHALLENGES

- 1. Ongoing flow of information → it is very obvious that the current setup of Worldvisitz lacks a lot of transparencies. XP has specific roles which would allow a more open communication
- 2. Agile XP Ceremonies - > In relation number 1, this would give a more structured approach for the continuity of information flow among the team and keeps everything organized. 2 Examples would be 'Pair Programming' & 'Refactoring'. The former would allow developers to further enhance the current code. The latter would enable the coders to restructure complex & poorly written code.
- 3. Agile (XP) roles- the roles using this framework are highly applicable to Worldvisitz' target goal. 2 examples would be the 'Tracker' & the 'Coach'. The first one focuses mostly on developers and also updates the team if he/she notices the process is slowing down. The second also acts as a 'Chief Architect' and also acts as a mentor to the less experienced members.
- 4. XP is non-traditional. It allows updates to happen whenever they are needed. It is non strictly conforming & always adaptable to changes. The roles and ceremonies would definitely prove this point.





BONUS: SCALING AGILE -> A QUICK 5 STEP OVERVIEW

- 1. Start with a Minimum Viable Product (MVP)
- This is very vital for early feedback. Also hypotheses could be derived based on the patterns of tracking usage. This step would save the wasted engineering time/efforts.
- 2. Creating a Product Backlog
- Product Managers and Product Owners should maintain 1 full backlog for the whole team. 1 specific backlog lets the Agile coach focus on top priority tasks while, while giving access to all possible contributors to that certain task.
- This step prevents miscommunication & further promotes collaboration& information sharing.
- 3. Create a Culture of Collaboration
- Highly recommended to have the Product Owner, Developer(s) & Tester(s) to have their specific meetings. PO expresses the business need, developer discussion architecture & implementation, tester examines potential issues, bottlenecks & other potential issues.
- 4. Using a Large Scale Framework
- Scaled Agile has 3 main frameworks such as: Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DAD) and Large Scale Scrum (LeSS). Each of these have their own specific uses cases and specialized areas wherein they could function better, and give maximum results.
- 5. Be a Lifelong Learner – it is very imperative for most team members to have a mindset of continuous learning. It pays a LOT (figuratively & literally) to have a keen interest or passion for this type of domain knowledge.