**Handy agile**

**Agile in Handy |Agile Coaching in Handy | Scrum in Handy | XP in Handy | Lean in Handy | Agile Planning in Handy |User Stories Estimation in Handy | Agile Metrics in Handy | Agile Soft Skills in Handy | Scaling Agile Framework**

SRIRAM

Dedication

Dedicated to Mr. Trivedi Jetty

Managing Director





|  |  |
| --- | --- |
| Sriram  I have been involved in IT Software development since 1997. I have unique combination of process, technical and industrial skills. As a Certified Agile Coach, I have expert level of knowledge in agile and practices with this combination I can help process and technology people, understand the agile world.  My agile coach journey started in 2011, when I was a part of Tata Consultancy Services. I practiced XP, scrum and agile methods thoroughly over several years and my teams are highly successful in delivering products using agile techniques.  I am proficient in agile engineering, coaching practices and SAFe consulting practices... Worked in USA, UK for TCS, Syntel Agile customers, which creates a global agile experience.  I have created agile websites and released books related to Scrum Alliance Professional, Agile Coaching and Agile A Key of Success & SAFe 4.0 Q&A.  As a Senior Agile Consultant helped to improve the teams of developers and overall project management by increasing collaboration, innovation, and introducing techniques that promote adaptability and flexibility in the face of unpredictable circumstances.  I am releasing “Handy Agile” to the Agile world made handy and expertise.  My sincere dedication to my agile guru’s with my agile journey. | Senior Agile Consultant    **My Core responsibilities:**   * Educate Colleagues * Mentor the Team * Develop Agile Adoption Strategy * Implement Agile Methodology * Agile Transformation |



|  |  |
| --- | --- |
| ***Message from Agile Guru*** | |
| Congrats again for your book release and website launch. We are proud of you .All the best. This is indeed achievement. | Wow! This is fantastic news! Congratulations!  All the best to you with your SPC4 Agile Coach career. |
| Regards, **Rakesh Khanna** Syntel Consulting Inc., | Interim CEO & President | Thanks,  **Susan Gibson**  Temenos LLC |CEO & Co Funder |

# **Agile in Handy**

|  |  |
| --- | --- |
| **Agile Intro**  **What is Agile?**  “Agile” is an umbrella term used to encompass dozens of different techniques and disciplines (e.g. Scrum, XP, KANBAN, etc.), all aimed at the iterative, incremental development of software.  The various disciplines | flavours of Agile are:-  **C:\Users\JS5027377\Desktop\agile um.png**  **What is the importance of Agile?**  The importance of the Agile are:-   * Reduce Turnaround time for features * Predictability of Market Releases content and thinking * Ability to handle complex product enhancements   **Why Agile?**  The main reason for the agile existence due to:-   * Project priorities change’s frequently * Need to respond to customer requirements and market dynamics * Promote team work and less reliance on individual heroics * Course correction and continuous improvements   **What is Agility? What does business agility mean?**  “Agility is the ability to deliver customer value while dealing with the inherent project unpredictability and dynamism by recognizing and adapting to change”. Agility comes with mobility.  **What is Agile Mindset?**  Image result for agile mindset | **What is Agile Manifesto? (Agile 4 Values)**  **- Individuals and interactions over Process and Tools** Self-organization and motivation are important, as are interactions like co-location and pair programming.  **- Working software over Comprehensive Documentation** Working software is more useful and welcome than just presenting documents to clients in meetings.  **- Customer collaboration over Contract Negotiation** Requirements cannot be fully collected at the beginning of the software development cycle, therefore continuous customer or stakeholder involvement is very important.  **- Responding to change over Following a plan** Agile methods are focused on quick responses to change and continuous development.  **What are the Agile Principles? (Agile 12 Principles)**  1. Customer Satisfaction 2. Welcome Changes  3. Frequent Delivery 4. Collocated Teams  5. Motivated Individuals 6. Face-to-Face Contact  7. Working Software 8. Constant Pace  9. Continuous Attention 10. Simplicity  11.Self-Organization 12.Regular Reflection  **What is need or demand for Agility?**  The demand of agility depends on the following factors:-   * Value-driven delivery * Faster time to market * Better transparency * Responsiveness to change * Self-managing teams * Sustainable pace of development * Reduced process waste * Better aligned with customer needs   **What are the key Agile practices?**  The key agile practices are:-   * Evidence based development – Empiricism * Limit work in progress ->Defer commitment | Match demand with capability | Progressive Elaboration | Adaptive Planning | Emergent Design * Visualize your work * Faster Feedback Cycle * Collaboration * Release early and often | Release on Demand * Continuous or Relentless Improvement |

|  |  |
| --- | --- |
| **Scrum**  **Tell me Scrum Framework in short?**  Agile has 4 Values & 12 Principles. SCRUM is a simple process framework.  SCRUM has:-  **3 Legs**: Inspect, Adapt, Transparent  **3 Roles**: Product Owner | Scrum Master | Development Team  **3 Artifacts**: Product Backlog | Sprint Backlog | Product Increment  **4 Meetings**: Sprint Planning | Daily SCRUM | Sprint Review | Sprint Retrospective  **1 Activity**: Product Backlog Refinement  **5 Values**: Focus | Courage | Openness | Commitment | Respect    **Product Backlog**-> Ordered list of items to be worked on for the product  **Sprint Backlog**-> Product backlog items selected to work in the Sprint and the work plan to complete those items  **Product Increment**-> Completed product backlog items in a sprint, which are ready to be delivered to the customer  **Product Backlog Refinement**-> A meeting to get the product backlog items ready for the next few sprints  **Sprint Planning**-> A meeting to create the sprint goal and plan the work for the sprint  **Daily SCRUM**-> A daily 15 minute time boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours  **Sprint Review**-> A meeting to inspect the product increment and adapt the product backlog if needed  **Sprint Retrospective**-> A meeting for the scrum team to inspect and adapt the process, people and tools  **XP**  **What are the overall flow of XP?**    **What is XP 5 values?**  XP 5 values are: Simplicity, Communication, Feedback, Respect & Courage  **What are the XP Practices?**    **What are the XP Concepts?**  **Test Driven Development->**Is a method that involves writing automated test code first and developing the least amount of code necessary to pass that test later  **Collective Ownership->**The entire team takes responsibility for the whole of the system  **Continuous Integration->**TheCode is integrated and tested many times a day, one set of changes at a time  **Metaphor ->**The metaphor consists of domain specific elements and shows their interconnectivity.  **Refactoring** -> Improve code quality without changing the behavior  **Pair Programming**-> In Pair programing, the entire code is written with two developers at one machine, with one keyboard and one mouse.  **Technical debt**-> Design and Coding, Imperfection that need correction  **Iterations** -> Design | Code | Test | Release within a specific duration  **Last Responsible moment**-> Delay till when it’s absolutely needed  **Stories**-> Self-contained elements taken up for implementation  **Time-boxing**-> Allocates time and wrap by that time  **Agile Planning**  **What are prioritization methods?**  The prioritization methods are:-   * MoSCoW Prioritization * 100 point method * Kano analysis * Relative weighting Model   **MoSCoW -** Primarily used in Dynamic System Development Method (DSDM)  **100 Point Method -** Each member given 100 points  Those item with the highest number of points is ranked first  **Kano Analysis –** Exciters or Delighters – Features which command extra price | Linear – More the better | Threshold – Must haves  **Relative Weighting Model –**   * Benefit of Presence & Penalty for absence is considered * Relative weighting (1 to 9 ) is provided by the P O’ s expert judgement * Desirability for each of the theme is calculated based on the relative weights   **What are the Earned Value Metrics?**  **Metrics Definition Calculation**  PPC Planned % Complete Iteration # / Total Iterations  APC Actual % Complete Points Delivered / Total Points  PV Planned Value PPC \* BAC  EV Earned Value APC \* BAC  CPI Cost Per Index EV / AC  SPI Schedule Per Index EV / PV  SV Schedule Variance EV – PV  CV Cost Variance EV - AC  NPV Net Present Value NPV = ∑ Ct / (1+r) t  **User Stories and Estimation**  **What is User Story and its format?**  User Stories provide a Light Weight approach to managing requirements for a system. Short statement of function captured in an index card or in a tool.  **Format of User Story**  **As a <role> I can <activity> so that <business value>**  E.g., As a user, I can login and gain access to the intranet, so that I can collaborate with all the organization  **What are 3 C’s of User Story?**  1.Card – Written on a card  2.Conversation –Details captured I conversations  3.Confirmation – Acceptance criteria confirm that the story is done  **What are types of User Story?**  **Requirements**  Functional requirements  Non-Functional requirements  **Technical User Story**  Compliance to standards and regulatory concerns  An item of work in the Software Debt backlog that pays off a chunk of debt that is undesirable and valuable to the Product Owner”  **Spikes Stories**  A story or task aimed at answering a question or gathering information, rather than implementing product features, user stories, or requirements.  Items needing investigation towards backlog grooming  **What is INVEST in User Story?**  **“Independent”**  The User story should be self-contained, in a way that there is no inherent dependency on another user story  **“Negotiable”**  User Stories, up until they are part of an iteration, can always be changed and rewritten  **“Valuable”**  A User Story must deliver value to the end user  **“Estimable”**  You must always be able to estimate the size of a user story  **“Scalable”**  User Story should not be so big as to become impossible to plan or Task or prioritize with some level of certainty  **“Testable”**  The User Story or its related description must provide the necessary information to make test development possible | **How Agile planning is done?**    **What are the Agile roles?**    **Explain the agile project management phases?**  **Envision** – Determine the product vision and project scope, the project community, and how the team will work together  **Speculate** – Develop a feature based release, milestone, and iteration plan to deliver on the vision  **Explore**– Deliver tested features in a short time frame, constantly seeking to reduce the risk and uncertainty of the project  **Adapt**– Review the delivered results, the current situation, and the team’s performance, and adapt as necessary  **Close**– Conclude the project, pass along key learning’s and celebrate  **Lean**  **What is the purpose of Lean?**  - Lean Development is focused on eliminating waste from a system and improving value to the customer. Lean is to minimize WIP  - Lean is for continuous improvement  **What are the Lean Principles & Practices**    **How will you eliminate waste in Lean?**  **What are the Value Stream Mapping Terms?**  **Value Stream Mapping**is a lean management method for analysing the current state and designing a future state for the series of events that take a product or service from its beginning through to the customer  **Lead time**is the time taken from when an issue is logged until work is completed on that issue. Lead time is what customer sees  **Cycle Time**is a measure of the time a work item takes to completeThe time a user story takes to get from the backlog to the done  **W I P(Work In Progress):**Number of work units in progress.  **Throughput**: The amount of material, data, work units that enters into a system and passes to generate output. Velocity, in Agile terms, can be the similar to this  **Little’s Law**: Cycle Time = W I P / A C R\* Where \*ACR = Average Completion Rate  **Total Cycle Time**= Value Added Time + Non Value Added Time  **Cycle Time Efficiency**: (Value Added Time / Total Cycle Time) \* 100  **What is Iteration Planning?**  In Iteration Planning – Identify the Iteration goal, Select User Stories, Split into Tasks & Estimate the tasks  Each story is taken, broken into tasks and estimated.  Team commitment is arrived at before including that within the iteration  The iteration activities are   * Be Specific, add all tasks (meetings, UT, bug fixing etc.,) * Add tasks of non-development nature-testing, documentation * Dependencies between user stories need to be handled * Task should not typically take more than a day   If tasks appear wrong, re-plan accordingly  **How will you estimate the velocity?**  **Use Historical Values**– From the similar teams under similar technology, domain, tool landscape and working environment  **Run an Iteration**- Run a few sample iterations and arrive at a narrow range of projected velocity per sprint. Typically 3-4 iterations will help to arrive at a projected value which is reasonably accurate range  **Make a Forecast -** When there is no scope for using historical values or sample sprints, make a forecast. Fill an iteration capacity with tasks derived from randomly selected stories. This can serve as indicative velocity  **What is velocity?**   * Velocity = User story points delivered in a sprint * Measure of work Team completes in a Sprint * Planning Tool * Only stories accepted by PO qualify for velocity calculation * Varies across teams * Projected Vs Actual Velocity   **Sprints SP Committed SP Accepted Velocity**  1 20 1616  2 24 24 24  **How will you split the user stories?**   * Can be split on data boundaries * Can be split alongside operations * Horizontal components * Functional and Non Functional Components * Maintain the INVEST check * Like “Shasimi” – Fish Slice   **What are the process metrics chart?**   * Sprint | Iteration Burn down chart * Release Burn down chart | Risk Burn down chart   **What is the difference between burnup & burndown?**   * Burndown chart shows how much work or effort remaining * Burn up chart shows how much work completed   **What do you think Velocity as Metrics?**  Nothing but the team capacity based on two factors mainly requirement ability & team capability  **What is Participatory Decision Model?**  Participatory decision models present different ways to engage the team in the decision making progress:-  - Mechanism in place for making decisions while still keeping everyone engaged in the project. - It is not realistic to expect the team to achieve total agreement on all issues and decisions  - These mechanism are called Participatory Decision Models  **1. Simple Voting**  A simple “For” or “Against” Voting  **2. Thumps Up | Down | Sideways**  Show of Thumps up, down or sideways around the room is a more efficient way of achieving a simple vote still allowing some time to discuss other options  Members holding the thumb sideways may be neutral on the idea, might have a conflict, concern or question that needs further investigation  **3. Jim Highsmith’s Decision Spectrum**  Team members indicate how they feel about a decision by placing a checkmark on the spectrum ranging from “Fully in Favor” to “Mixed Feelings” to “Absolutely No” or Veto  It could be created on a whiteboard with permanent markers  **4. Fist of Five**  People vote by showing the number of fingers that indicates their degree of support  One Method: A first (No Fingers) means to support, while raising five fingers indicates full support  Other Method: (Most Popular)  **One Finger:** I totally support this option  **Two Fingers:** I support this option with some minor reservations that we probably don’t need to discuss  **Three Fingers:** I have concerns that we need to discuss  **Four Fingers:** I object and wants to discuss the issue  **Five Fingers:** Stop, I am against this decision |

# **Agile Coaching in Handy**

|  |  |
| --- | --- |
| **What is Coaching?**  Coaching is partnering with clients in a thought-provoking and creative process that inspires them to maximize their personal and Professional Potential.  *Coaching is a form of development in which a person called a coach supports a learner or client in achieving a specific personal or professional goal.*  **What is Coaching?**   * Facilitate positive change * Explore and enhance individual/team potential * Enable self-learning and self-growth   **How to enable Self-Learning and Self-Growth?**   * Make them think * Engage their conscious mind * Asking powerful questions is the primary tool of a Coach!   **What are Powerful Questions?**  Powerful Questions that:-   * Generate insight in Coachee’s mind * Motivate Coachee for action * Invoke commitment   **How to handle teams who are resistant to move into Agile?**  To handle the team move into the agile:-   * Based on Agile Mind-set and Culture * Team Maturity   **Why worry about change?**  The main reason for the change is to:-   * Market dynamics * Client expectations * Technology * Competition   **What motivates people for change?**  **Away from Change:** Fear of failure | Comfort Zone | Lack of learning experience  **Towards Change:** Struck in Something | Breaking the Status quo  **What SCARF® Model?**   * **Status** is about relative importance to others. * **Certainty** concerns being able to predict the future. * **Autonomy** provides a sense of control over events. * **Relatedness** is a sense of safety with others - of friend rather than foe. * **Fairness** is a perception of fair exchanges between people   There are also different behavioral and psychological consequences associated with threat and reward:-  **Threat leads to:**  - Reduced working memory | Narrower field of view | Generalizing of threat | Greater pessimism  **Reward leads to:**  - Greater cognitive resources | More insights | Increased ideas for action  - Fewer perceptual errors | Wider field of view  **How will you cultivate growth Mind-set?**  o Facilitate Learning goals  – Mastery and Competence  o Metrics reflect current state (NOW)  – Measuring potential or productivity is a lower priority  o Focus on positive emotion  – Performance and Enjoyment – Decrease negative emotion  **What is Servant Leadership?**  The term coined by Robert K. Greenleaf:  o Lead by serving others  o Provide leadership in helping others achieve their goals:  o Focus on building capacity in others  o Facilitate decision making process  o Distance from the center of action and attention  o Listen and give space  **Tell me the meaning of Japanese Term “Shu Ha RI”?**  **Shu** - Learn exactly what they taught by master  **Ha** - Experimental Stage (Creating own pattern better than master)  **RI** - Doing Agile and Being Agile  **What are the ICF Coaching Competencies?**  Coaching is about behavioral competencies:  **Setting the Foundation**1. Meeting Ethical Guidelines and Professional Standards 2. Establishing the Coaching Agreement  **Co-creating the Relationship**3. Establishing Trust and Intimacy with the Client 4. Coaching Presence  **Communicating Effectively**5. Active Listening 6. Powerful Questioning 7. Direct Communication  **Facilitating Learning and Results**8. Creating Awareness 9. Designing Actions 10. Planning and Goal Setting 11. Managing Progress and Accountability  **Why transition to Agile?**  Agile creates wonders with every life nowadays. more easy adaption , global model, good imagination clients thoughts get reflected immediately, provides good leadership level for organization to support customers, as an entrepreneurial customers, it shows the value of an organization at higher level  **Essence of Scrum**– Quicker Value Realization – Empower & Motivated Teams – Customer Satisfaction – Quality  **Essence of Kanban**– Better Visualization – Faster Throughput, Higher Cyclic Team – No Team limit, Allows Specialist Roles  **What are the specific points to remember during agile coaching?**  The specific points to remember in agile Coaching are:-   * **Is Agile training part of the engagement?**   - How often will the agile training will be conducted?  - How long will each session be?   * **How many locations will be covered in coaching?** * **How many teams are part of it?** * **Will coaching involve working with Scrum Masters and/or the entire team?**   **Professional Coaching Skills**  **What is coaching presence? What is ICF?**  “Ability to be fully conscious and create spontaneous relationship with the coachee By employing a style that is open, flexible and confident.” – I C F means International Coach Federation  **What is key motivation for coaching presence?**  The key motivations of Coaching presence:-  - Build rapport with the coachee  - Engage coachee in the coaching process  - Keep the coachee in ‘towards’ state  **What is Active Listening? What are its approach?**  “Ability to focus completely on what the client is saying and is not saying, to understand the meaning of what is said in the context of client’s desires, and to support client’s self-expression.” - ICF  **Attend to the client’s agenda**  **Listen to client at all levels**  - words, tone of voice and the body language  - feelings, perceptions, concerns, values and beliefs  - concerns, goals, ideas, suggestions  **Help the client get there**  - Focus on the essence of client's communication  - Allow the client to vent without judgment or attachment, bypass long and descriptive stories  - Mirror back what client has said – summarize, paraphrase, reiterate  - Integrate and build on client's ideas and suggestions  **What is direct communication? Why it is so important?**  “Ability to communicate effectively during coaching sessions, and to use language that has the greatest positive impact on the client.” – ICF  - Clearly state coaching objectives and agenda  - Speak with intent  - Be clear, articulate and direct in sharing and providing feedback  - Reframe and articulate  - Help the client understand from another perspective  - Use appropriate and respectful language  - Non-sexist, non-racist, non-technical, non-jargon  -Use metaphor and analogy  **Tips for giving feedback?**  The tips for giving feedback to the team are:  - Listen carefully  - Give timely feedback  -Think before you speak  - Observation vs. Evaluation  -Be succinct, specific and generous  -Always ask permission to touch sensitive areas  - Balance positives and negatives  **Why you give feedback to the team?**  Effective feedback helps keep the conversation meaningful:  - Closes the communication loop  - Confirms the listener has understood what is being said  - Assists in highlighting gaps or assumptions in a timely manner  **Define Emotional Intelligence? What are its performance?**  Emotional intelligence is the habitual practice of:  - Using emotional information from ourselves and other people And, integrating this with our thinking:  - Better decision making in defining life goals and problem solving  **Performance**  - Improves relationships  - Improves confidence and positivity  - Improves career prospects  **What are Multiple Intelligence’s?**  Howard Gardner’s research into multiple intelligences:  An ability to recognize and understand other people’s moods, desires, motivations, and intentions  1. Linguistic | 2. Logical – Mathematical  3. Spatial | 4. Musical |5. Naturalist |6. Bodily – Kinesthetic  7. Intrapersonal |8. Interpersonal  **Coaching Conversations**  **How would you improve the coaching performance?**  To improve the performance of coaching, the following activities to be performed:-  1. Think about thinking  2. Listen for potential  3. Speak with Intent  4. Dance towards insight  5. CREATE New thinking  6. Follow-up  **What is Think about Thinking?**  1. Let them do all the Thinking  2. Focus on Solutions  3. Remember to Stretch  4. Accentuate the Positive  5. Put Process before Content  **What do you think Listen for Potential?**  Coaching is about:  - Enabling self-learning and growth  - Unlocking potential to maximize performance  **Why do we speak with intent?**  Speak with focus on maximizing improvement in thinking:  1. Succinct, 2. Specific, 3. Generous  **What are the principles of Collaboration?**  -Help promote high degree of collaboration  - With Strong result orientation  **What are the characteristics of Collaborative teams?**  -They are self-organizing rather than role- or title-based.  -They are empowered to make decisions.  -They truly believe that as a team they can solve any problem.  -They are committed to team success vs. success at any cost.  - The team owns its decisions and commitments.  - They are consensus-driven, with full divergence and then convergence.  - And they live in a world of constant constructive disagreement  **What are the Five Dysfunctions of a Team?**  Absence of Trust, Fear of Conflict, Lack of Commitment, Avoidance of Accountability & Inattention to Results  **How Collaboration demands behavior changes?**  **Non Collaborative**  Individual Accountability  Rely on managerial control  Divide work into tasks and assign to people  Build Functional Excellence by narrowing set of tasks done by individual  Maximize individual productivity - A fair day’s pay for fair day’s work  **Collaborative**  Joint Accountability in addition to Individual accountability  Encourage people to manage their own work  Expect everyone to plan work together and pull tasks  In an effort to finish work faster, encourage people to learn and volunteer for work outside their specialization  Support personal growth that expands as well as exploits individual capabilities  **How will you manage the distributed teams?**  **Favor live communication over email and documentation**  **Create opportunities to bring the team together**  -Initial co-location  - Seeding Visits  - Contact Visits  - Traveling Ambassadors  **Adding Some Process and Structure**  - Agree on one common language  - Clarify tasks and processes, not just goals and roles  - Use of collaboration tools like Jira, Wiki, SharePoint, etc.  - Insist on whole team presence in team meetings | **What are all the Agile Coaching Roles?**  **Teaching:** Share knowledge to help them understand agile values, principles, and their role  **Mentoring:** Show and help them practice how work gets done in an agile process  **Coaching:** Help them find solution to their problems on their own, in line with Agile values and principles  **What are the focus areas or goal of an agile coach?**   * Train them on Agile Framework * Develop Agile Mindset * Develop Lean Mindset * Develop T-Skilled Professionals   **What are the benefits or ground work for one-on-one-coaching?**   * Guarantee Safety * Partner with Manager * Create a positive regard * Meet them a half-step ahead   **How will you coach the product owner based on the role?**   * Business Value Driver * Vision keeper * Daily Decision maker * Heat shield * One ultimately responsible   **What are the key tasks in coaching the Product Owner? How will you support PO?**  Mentor Product Owners understand their priorities:   * Be the vision keeper – in sync with sponsor * Move from schedule-driven to business-value-driven planning * Cultivate Business value driven thinking in all interactions * Maintain a DEEP product backlog * Match demand with capability   - Learn to trust the team  - Avoid micro-management   * Hold the team for their commitments   **How will you coach the Scrum Master based on the role?**   * Caretaker of the agile process * Servant leader * Facilitator * Progress tracker towards the goal * Bulldozer for impediments * Guardian of quality   **What are the key tasks in coaching the Scrum Master?**  **How will you support SM?**  Coach Scrum Master understand their priorities:   * Facilitator, not decision maker * Self-managing teams   - Planning work  - Pulling work  - Tracking work  - Getting work done   * Sustainable pace of development   **Map the resistance stickie’s to different SCARF categories?**  **Team**  **New Process** –**Certainty More Transparency & Accountability** -**Autonomy Knowledge Sharing** – **Autonomy Colocation** – **Autonomy | Certainty**  **Team Lead/Team Manager / Delivery Manager**  **Dynamic Planning** – **Certainty Roles Change** – **Status Self-Managing Teams** – **Autonomy Customer Collaboration** – **Autonomy**  **Senior Management**  **Empowerment** – **Autonomy Transparency to Customer** – **Autonomy Variability in Productivity** –**Certainty**  **Customer**  **Time Availability** – Autonomy **Contract Model Changes** – Certainty  **Coaching Agreement**  **What is coaching stance?**  The Coaching discipline are:  o Holding the client’s agenda  o Maintaining neutrality  o Reducing client dependence  o No colluding  o Self-awareness and self-management  o Signature Presence  **What are the challenges of internal coach?**  The Challenges of internal coach are:  o Maintaining neutrality and confidentiality  o Holding people accountable  o Challenging the status quo  o Personal bias  o Difficult to stay out of office politics  **What are the challenges of external coach?**  The Challenges of external coach are:  o Need for effective system entry  o Treating the client process with respect  o Balancing challenge and pace of change  o Understanding organization culture is a slow process  o Fostering too many new relationships  **What are the coaching pre-requisite?**  The Coaching pre-requisite are:  **Purpose**  Does the coachee /team have a clear goals that focus on delivering the business value?  **Commitment**  Is the coachee / team committed to achieve their goal?  **Motivation**  Is the coachee / team motivated to work with the Coach?  **Expectations**  Does the client / coachee have realistic expectations from the Coach?  **Investment**  Is the client / coachee / team ready to invest time, money and energy in the coaching process?  **What is coach must not do? What Project Manager do**  Coach has to adapt the Coaching discipline not the PM role:  -Manage teams or team members  -Give performance evaluations for team members  - Track Delivery issues  -Work with clients in contract negotiations  **What is powerful questioning?**  “Ability to ask questions that reveal the information needed for maximum benefit to the coaching relationship and the client.” - ICF  **Asks questions that reflect active listening**  - Reflect understanding of the client's perspective  - Challenge client's assumptions  - Facilitate new insight, discovery or possibility  - Prompt commitment to action  **Asks questions that move the client forward:**  -Towards goals that they desire  - Not backwards to problems and justifications  **What are the questions likely to generate Insight..?**  The questions likely to generate Insight because:-  **Forcing people into action**  What is your next step…? What are you doing about…?  **Telling people how to think**  Why don’t you think about…? What about being more positive... **Asking “why” questions to get more details**  Why is this issue…? Why did you... Why don’t you…?  **Giving Advice**  Why don’t you… What about doing this…  **Asking questions about the problem**  What is the issue…? What is stopping you…? Why do you think this…?  **Asking questions to get more details**  Can you list….? What’s been happening …?  **What are the questions that generate Insight..?**  The questions that generate Insight are:-  **Thinking Questions**  How long have you been thinking about…? What are your insights so far? What ideas have crossed your mind about…?  **Vision questions**  What would you like to achieve here? What is your objective, goal or desired outcome?  **Planning Questions**  What are the milestones for achieving this goal? What are your options?  **Action Questions**  What are you prepared to do for…? What will you complete by next week?  **Labelling questions**  What are three key emotions around…? If you could label your emotion, what would it be?  **Reappraisal Questions**  How might you handle this differently? How could you think from a different perspective?  **What are the four faces of Insight?**  The four faces of Insight are:-  1. Awareness of Dilemma  2. Reflection  3. Illumination – the ‘Aha!’ moment  4. Motivation  **What is Dance of Insight?**  A coaching model that helps:-  - Helping people develop a deeper awareness of their dilemma  - Keeping people fully engaged towards making new connections  - Making the ‘Aha!’ moments happen  The Dace of Insight Model is  1. Permission2. Placement  3. Questioning4. Clarifying  **What is Grow Model? How it suits for coaching?**  Grow model that helps in coaching:-  G-Goal | R-Reading | O-Options | W-Will (What | When)  **What is Create Model? How it suits for coaching?**  The idea of CREATE model is to ‘Coach for Action’:  - Current Reality  - Explore Alternatives  - Tap their Energy  **What is Follow up?**  We can follow up with the following list of items:-  - Facts , Emotions, Encouragement, Learning , Implications &  New goals  **Team Development**  **What is a team?**  “A team is a small number of people with complementary skills who are committed to a common purpose, performance goals and approach for which they hold themselves mutually accountable”  **What are the different teams? How helping a team choose to be a team?**  -Working Group  - Pseudo or Potential Team  - Real Team  - High Performance Team  **Working group**  Strong, clearly focused leader  Individual accountability  The group purpose is the same as the broader organization mission  Individual work products  Meetings focus on status check  Measure its effectiveness indirectly by its influence on others  **Pseudo or Potential Team**  A history of working together  -A significant, incremental performance need  - Requires more clarity about common purpose and goals  - Lacks a disciplined common working approach  - The practice of collective accountability is not established yet  **Real team**  Shared Leadership role  Individual and shared accountability  Specific Team purpose that team itself delivers  Collective work products  Meetings focus on open-ended discussions and active problem solving  Measure performance directly by assessing collective work products  **High performance agile team**  Agile teams aim to be high performance teams in house frame:  - Propelled by shared purpose  - **High Trust**as a **base**& Aligned with **roof level**as a “Value”  **4 Pillars**are  -Shared **leadership**  Free from Command and Control culture  - High degree of **collaboration**  - **Continuously seeking to improve**  - **Self-managing**  **What are the 5 stages of Team Development?**  Forming | Storming | Norming | - Performing | Adjourning  **What is groupthink?**  “A phenomenon that occurs when the desire for harmony or conformity in the group results in an irrational or dysfunctional decision-making outcome.”  **What are the responsibilities of agile coach in handling conflicts?**  - Skillfully determines the severity of conflict  - Mindfully decides whether to intervene and how  - Generously teaches teams how to navigate it  - Courageously refuses to settle for a team that tries to hide or avoid it  **What are the Five Levels of Conflict?**  Problem to Solve, Disagreement, Contest , Crusade , World War  **How would you respond to conflict?**    **Setting up the Team Environment**  **What are the challenges with distributed agile teams?**  The challenges in distributed agile team are:-  - **Communication**  Limited or no F2F communication  - **Collaboration**  Difficult to have the entire team together  - **Transparency**  Work assignment and status is not visible across locations  - **Shared Learning**  Difficult to share knowledge across locations  - **One Team Feeling**  Common to have development of silos – We Vs. Them  **How will you support the Product owner to handle the product backlog?**  As a coach I support the product owner to guide them to:-   * Maintain a DEEP product backlog * Focus on business-value-driven * Using MoSCoW technique to maintain the priority |
|  |  |

# **Scrum in Handy**

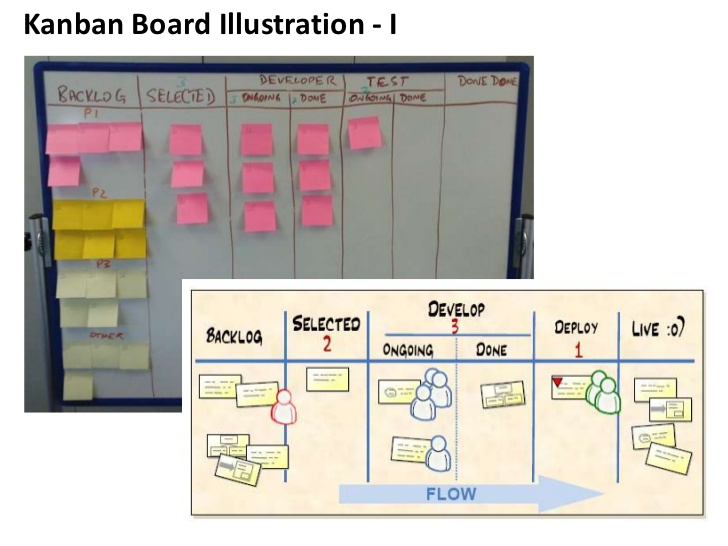
|  |  |
| --- | --- |
| **What is SCRUM?**  A framework within which people can address complex and adaptive problems while productively and creatively delivering products of the highest possible value.  **What is the benefit in using SCRUM?**  An agile Scrum process benefits the organization by helping it to:-   * Increase the quality of the deliverable’s * Cope better with change (and expect the changes) * Provide better estimates while spending less time creating them * Be more in control of the project schedule and state   **Why use Scrum?**  The Key benefits of using Scrum in any project is:-   * Adaptability, Transparency, Continuous Feedback, * Continuous Delivery of Value, Sustainable Pace, * Early delivery of high value, Efficient Development Process * Motivation, Fail Fast, Faster Problem Resolution * Effective Deliverable's, Customer Centric * High Trust Environment, Collective Ownership * High Velocity, Innovative Environment   **What are the characteristics of Scrum?**   * The most popular ‘Agile Process’ for software development * A project management/execution process framework * Well suited for projects that require Empirical process control * Focuses on self-organizing teams * Requirements are captured in a prioritized list (Product Backlog) * Product progresses in a series of month-long “sprints” * No specific engineering practices prescribed   **What are the Scrum Principles?**  SCRUM principles are the core guidelines for applying the Scrum framework and should mandatory be used in all Scrum projects. The six Scrum principles are:-   * Empirical Process Control * Self-organization * Collaboration * Value-based Prioritization – Product based progress * Time-boxing – Delivering Projects * Iterative Development   **What are Scrum Processes?**  Scrum processes address the specific activities and flow of a Scrum project. In total there are nineteen processes which are grouped into five phases such as:-   * Initiate, * Plan and Estimate * Implement * Review and Retrospect * Release   **Scrum Roles**  o Product Owner  o Scrum Master  o Development Team  **What are the responsibilities of the Product Owner?**   * Defines the feature of the product * Decide on release date and content * Be responsible for the profitability of the product (ROI) * Prioritize features according to the market value * Adjust features and priority every iteration, as needed * Accept or reject work results * Maintains and grooms the Product Backlog * An effective product owner is Committed, Responsible, Authorized, Collaborative, and Knowledgeable (CRACK)   **What are the responsibilities of the Scrum Master?**   * Represents management to the project * Responsible for enacting Scrum rules, values and practices and censure team members and stakeholders not adhering to these rules ad norms * Removes impediments * Ensure that the team is fully functional and productive * Enable close co-operation across all roles and functions * Shield the team from external interferences * Practices “Servant Leadership” – Facilitator and enabler rather than a Manager   **What is the Scrum team size? How about Scalability in Scrum?**   * Typically individual team is 7 +- 2 people * Scalability comes from teams of teams * Factors in scaling   + Type of application   + Team Size   + Team dispersion   + Project duration   **What is Product vision?**  Product vision statement is few sentences that talks about the motivation behind the product. Since the product owner is responsible for success of the product, he/she leads the creation of Product Vision.  For Creating the Product Vision   * Describe the Motivation behind the Product * Look beyond the Product * Distinguish between Vision and Product Strategy * Employ a Shared Vision * Choose an Inspiring Vision * Think Big * Keep your Vision Short and Sweet * Use the Vision to Guide your Decisions   **What is Start | Stop | Continue?**  Whole team gathers and discusses what they would like to do and Product manager has the authority to do.  **What is Sprint Retrospective?**  A meeting for the scrum team to inspect and adapt the process, people and tools   * Periodically take a look at what is and is not working * Typically 15 – 30 minutes * Done after every sprint * Scrum Master facilitates and team participates. Potent tool for improvement -> Sprint after Sprint * Whole team participates * Scrum Master * Product Owner (Optional) * Team * Others (Not desired)   **Scrum Artifacts**   * Product Backlog * Sprint Backlog * Burndown Chart   **What is Product Backlog?**  Ordered list of items to be worked on for the product .The requirements are:   * A list of all desired work on the project * Ideally expressed such that each item has value to the users or customers of the product * Prioritized by the Product Owner * Reprioritized at the start of each sprint   **What is Product Increment?**  Completed product backlog items in a sprint, which are ready to be delivered to the customer with   * High Quality * Perfectly Tested * Completed Scope * Ready to Use * As per Definition of Done   **What is Product Backlog Refinement?**  A meeting to get the product backlog items ready for the next few sprints  **What is Sprint Goal?**  A short statement of what the work will be focused on during the sprint  **What is Sprint Backlog?**  Product backlog items selected to work in the Sprint and the work plan to complete those items   * Individuals sign up for work of their own choosing – Work is never assigned * Estimated work remaining is updated daily * Any team member can add, delete or change the sprint backlog * Work for the Sprint changes * If work is unclear, define a sprint backlog item with a larger amount of time and break it down later * Update work remaining as more becomes known   **What is Definition of Done?**  **Definition of Done is a list of attractive activities agreed by the Product Owner and the Development Team to call a backlog item is done**   * Definition of Done consist of activities needed for functional and quality requirements * Team comes up with the DOD and adheres to it while creating the product increment * Different teams may have different DOD but all teams should follow minimal DOD that includes all critical activities required * If there are standards at organizational level, a common DOD can capture those and the teams should have separate DOD in addition to one at the organizational level * A **Stronger DOD** leads to higher quality product: - * Code Complete * Unit tests Written * Code Review * Manual Functional Testing * Automation * Updated Documents * User Acceptance Testing * Successful Deployment * Suppose if the DOD is missing essential activities, it is called a Weak DOD. For E.g. Load testing may not be done for every sprint and is deferred to later time. * A **Weak DOD** causes unfinished work in every sprint. The unfinished work is added back to product backlog. This increased the risk. If a major bug is found during the load testing, that could risk the release * Weak DOD also **increases the Technical Debt**. This might include the automation or code reviews | **Tell me Scrum Framework in short?**  Agile has 4 Values & 12 Principles. SCRUM is a simple process framework.  SCRUM has:-  **3 Legs**: Inspect, Adapt, Transparent  **3 Roles**: Product Owner | Scrum Master | Development Team  **3 Artifacts**: Product Backlog | Sprint Backlog | Product Increment  **4 Meetings**: Sprint Planning | Daily SCRUM | Sprint Review | Sprint Retrospective  **1 Activity**: Product Backlog Refinement  **5 Values**: Focus | Courage | Openness | Commitment | Respect    **Product Backlog**-> Ordered list of items to be worked on for the product  **Sprint Backlog**-> Product backlog items selected to work in the Sprint and the work plan to complete those items  **Product Increment**-> Completed product backlog items in a sprint, which are ready to be delivered to the customer  **Product Backlog Refinement**-> A meeting to get the product backlog items ready for the next few sprints  **Sprint Planning**-> A meeting to create the sprint goal and plan the work for the sprint  **Daily SCRUM**-> A daily 15 minute time boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours  **Sprint Review**-> A meeting to inspect the product increment and adapt the product backlog if needed  **Sprint Retrospective**-> A meeting for the scrum team to inspect and adapt the process, people and tools  **Scrum Ceremonies | Meetings**   * Sprint Planning * Sprint Review * Sprint Retrospective * Daily Scrum Meeting   **What is Sprint Planning? What are all the inputs and outputs of Sprint Planning?**  A meeting to create the sprint goal and plan the work for the sprint.   * Product Owner and team negotiates stories to select in current sprint from prioritized product backlog. * Selected stories are estimated with agreed acceptance criteria * Team identifies and estimates Task * Scrum Master, Product Owner and team can attend the meeting * Product Backlog has to be groomed by PO prior to meeting * Normally split into 2 sets of 4 hours each * First half for choosing the product backlog items with PO * Second half for splitting into tasks and assignment * Product Owner is optional for the second half * Inputs - Team Capacity, Marketing Backlog, Business Priorities, Product Status, Competition * Artifacts – Sprint Goal , Sprint Backlog (Output of Sprint Planning)   **What is Daily Scrum?**  A daily 15 minute time boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours:-   * Parameters * Daily for 15 minutes * Typically first thing is the day * Stand-up * At same place * Not for problem solving * Whole world is invited * Only team members, Scrum Master, Product Owner can talk, Pigs talk, Chickens observe * Helps avoid other unnecessary meetings * Scrum Master to intervene to bring in discipline after due attempts at self-correction * Every one answers 3 questions * What did you do yesterday? * What will you do today? * Is anything in your way * These are not status for the Scrum Master – They are committed in front of the peers   **What is Sprint Review?**  A meeting to inspect the product increment and adapt the product backlog if needed   * Team presents what it accomplished during the sprint * Typically takes the form of a demo of new features or underlying architecture. Done from QA server (Close to Prod setup) * Informal * Time boxed to 4 hours | 2 hour prep time rule * No slides. Artifacts shown on need * Whole team participates * Invite the world * Only “Done” stories are demonstrated * Stakeholders can provide comments which go into the Product Backlog   **What is Sprint Burndown Chart?**   * Primary method of tracking progress * A Burn down chart shows how much work is left as of a date * Scrum Master encourages team to use the Burn down chart as guidance in managing the sprint work   **What is Release Planning?**  The release planning is a tentative plan for the whole release that covers several sprints  The Input for release plans are –   * Release strategy, Priority * Estimated Product backlog that gives the total size of the release * Velocity of the team which represents the productivity * Assumptions, constraints and risks   **Total size of the backlog**   * Development team estimates items in the backlog. Since all items won’t be clear. The team makes their best guess. * Any estimation technique like planning poker or affinity estimation could be used * Team can iterate over estimates until they feel that the overall estimate is roughly accurate   **Velocity**   * A long term measure that indicates how much work is “done” per sprint * Velocity is number of point completed per sprint * Particularly finished stories don’t count * Velocity varies in every sprint   **How do we know the team’s velocity?**   * If the team is in place for some time, look at the history of the team’s velocity * If the team is new, run couple of sprints to establish the initial velocity * Use the average velocity over several sprints to predict the completion date * Let’s plan a release   Let’s say the release backlog of size 100 points. The team ran 4 sprints and had velocities of 8, 11, 9, 10  Average velocity for 4 sprints: 38/4 = 9.5  Best Velocity: 11 | Worst Velocity: 8  Remaining points after 4 sprints: 62 | Required sprints to complete 62 points (Round up value)  @ Average velocity: 62/9.5 = 7 sprints |  @Best Velocity: 62/11 = 6 sprints |  @ Worst Velocity: 62/8 = 8 sprints  The output for release plans are –   * Release scope/date * Prioritized backlog * Ordered estimated in size and forecasted in sprint, Release burnt chart   **Identify Scrum Ownership | Responsibility?**  **Items Responsibility**  Estimates Development Team  Velocity Predictions Development Team  Technical Decisions Development Team  Process Adherence Scrum Master  Agile Coaching Scrum Master  Backlog Priorities Product Owner  D O D Development Team Product Owner Scrum Master  Sprint Planning Development Team Product Owner Scrum Master  **What is Scrum of Scrum?**  A technique to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of 5-10.  Each daily scrum within a sub-team ends by designating one member as **"Ambassador"** to participate in a daily meeting with ambassadors from other teams, called the Scrum of Scrums.  The Scrum of Scrums proceeds otherwise as a normal daily meeting, with ambassadors reporting completions, next steps and impediments on behalf of the teams they represent.  Resolution of impediments is expected to focus on the challenges of coordination between the teams; solutions may entail agreeing to interfaces between teams, negotiating responsibility boundaries, etc.  The Scrum of Scrum will track these items via a backlog of its own, where each item contributes to improving between-team coordination.  Also known as a **"Meta Scrum".**  **Discuss Scrum Positive and Challenges?**  **Positive**  $ Inspect | Adapt | Transparency $ Fail fast | Faster feedback  $ Collective ownership $ Continuous Improvement  $ Self Organized team $ Collaboration $ Engagement  $ Self-Delivery Teams $ Well defined roles | ceremonies  $ Early Feedback $ Accountability $ Shared risk  $ High Quality $ Excellent Productivity  **Challenges**  $Skill set for co-locations  $ Delivery commitment of every two weeks  $ Not having deliverable after every Sprint  $ Larger Team $ Mind-set Change  $ Ad-hoc requirements within the Sprint $ Resource Criticality  $ Inefficient resource $ Adaptability & Sustainability  $ Time-box collaboration $ Team level limit (Only for smaller teams)  $ Cross functional team structure $ Framing Agile centric metrics  $ Sometimes story point estimation |

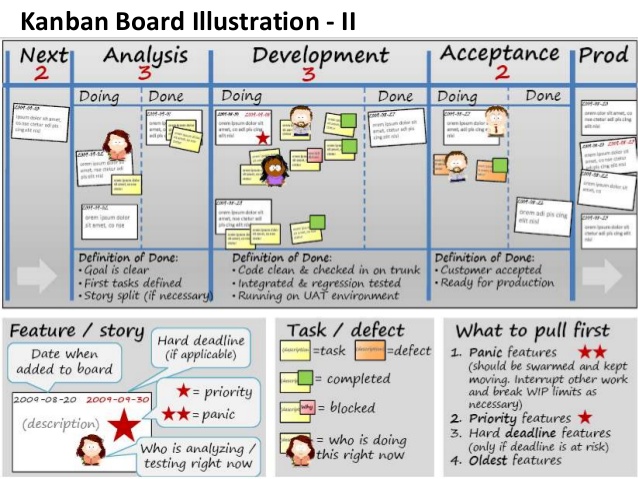
# **XP in Handy**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **What is XP?**  XP recognizes that:   * All requirements will not be known at the beginning * Requirements will change * Use tools to accommodate, change as a natural process Do the simplest thing that possibly work and refactor * Emphasis values and principles rather than process   **What are the XP Practices?**    **What are the XP Concepts?**  The XP concepts are:-  **Test Driven Development ->**Is a method that involves writing automated test code first and developing the least amount of code necessary to pass that test later  **Collective Ownership ->**The entire team takes responsibility for the whole of the system  **Continuous Integration ->**The Code is integrated and tested many times a day, one set of changes at a time  **Metaphor ->**The metaphor consists of domain specific elements and shows their interconnectivity. Analogy to explain things or term used to remember a thing  **Refactoring** -> Improve code quality without changing the behavior  **Pair Programming**-> In Pair programing, the entire code is written with two developers at one machine, with one keyboard and one mouse.  **Technical debt**-> Design and Coding, Imperfection that need correction  **Iterations** -> Design | Code | Test | Release within a specific duration  **Last Responsible moment**-> Delay till when it’s absolutely needed  **Stories** -> Self-contained elements taken up for implementation  **Time-boxing**-> Allocates time and wrap by that time  **XP Roles – Onsite Customers**  **What they do**   * Release Planning * Provide Requirement details and answer queries * Prioritizing Stories * Reviewing work in progress * Leading Iteration demos * Stay one step ahead of developers-Crunching user stories and grooming the backlog train for feature iterations   **Who**   * Product Manager * Domain Expert * Interaction Designer * Business Analyst   **XP Roles – Programmers**  **What they do**   * Pair Programming * Incrementally designed and architect * Pare down technical debt * Configuration Mgmt.: Single core base and automated build * Implement code and refractory * Adhere to coding standards   **Who**   * Programmers * Designers * Architects * Technical Specialist   **XP Roles – Testers**  **What they do**   * Automate regression test * Test for functional and nonfunctional requirements * Exploratory testing * Automated test in nightly build   **Who**   * Manual tester * Automation Engineer   **XP Roles – Coach**  **What they do**   * Process coaching * Technical coaching * Access to resources * Impediment removal * Guidance and support   **Who**   * Coaches * Project Managers * Quality analysts * Executive Sponsors | **What are XP 5 values?**  XP 5 values are: Simplicity, Communication, Feedback, Respect & Courage  **What are the overall flow of XP?**    **What are the XP Practices?**   |  |  |  |  | | --- | --- | --- | --- | | **Thinking** | **Collaborating** | **Releasing** | **Developing** | | **.**Pair Programming  **.**Energized workspace  **.**Root Cause Analysis  **.**Informative Workspace  **.**Retrospectives | **.**Trust  Sitting together  **.**Real customer involvement  **.**Metaphor  **.**Standup Meetings  **.**Coding Standards  **.**Iteration demo | **.**“Done”  **.**  No Bugs  **.**Version Control  **.**10 minute build  **.**Continuous Integration  **.**Collective code ownership  **.**Documentation | **.**Incremental requirements  **.**Customer test  **.**Test driven development  **.**Refactoring  **.**Simple design  **.**Incremental design and architecture  **.**Spike Solutions  **.**Performance optimization  **.**Exploratory testing |   **Difference between Scrum and XP?**   |  |  |  | | --- | --- | --- | | **Scrum** | **XP** | **Definition** | | Sprint | Iteration | Fixed length period of time | | Release | Small release | Release to production | | Sprint | Release Planning | Planning game | Agile Planning meetings | | Product Owner | Customer | Business representative to projects | | Retrospective | Reflection | Lesson Learned Style meeting | | Scrum Master | Coach | Agile Project Manager | | Development Team | Team | Empowered cross functional team | | Daily Scrum | Daily Standup | Brief daily status meeting |   **What is technical debt?**  Technical Debt is the cost of shortcuts (like hard coding, not following coding standards) that accumulate over time, leading to:   * Increased risk * Slower time to market * Greater maintenance and enhancement costs * Unmanaged Technical Deb leads to systems too unwieldy to use, and too expensive to fix |

# **Lean Kanban in Handy**

|  |  |
| --- | --- |
| **What is Lean Software Development?**  **What is the purpose of Lean?**   * Lean Software Development (LSD) is a translation of Lean manufacturing and lean IT principles and practices to the software development domain taken from Toyota Production System (TPS) by Taiichi Ohno * Lean is most popular with start-ups that want to penetrate the market, or test their idea and see if it would make a viable business * **Lean Development is focussed on eliminating waste from a system and improving value to the customer. Lean is to minimize WIP** * Lean creates process speed up by eliminating waste. Lean improves efficiency & quality by eliminating waste**. Lean is for continuous improvement** * Lean principles can be applied to any business context and applicable to a wider scope * Lean has no formal methodologies, as against in Agile. In fact, Lean is Not an Agile methodology. Agile was developed on Lean Principles * Lean views specific Agile Software Development methodologies, as supporting practices * Adapted by Mary and Tom Poppendieck – Lean Software Development Kit: An Agile Toolkit   Source Website: http://www.poppendieck.com  Lean aims for DEBASED wastes:-   * Defer Commitments * Empower team * Build quality in * Amplify Learning * See the Whole * Eliminate Waste * Deliver Fast * Respect People * DOTWIMP     **What are the 7 Wastes per Lean?**  7 Wastes per lean : D O T W I M P   * Defects * Overproduction * Waiting * Inventory * Motion * Processing (Extra)   **What is 5S?**  5 S is a methodology that had come out of the techniques within Total Productive Maintenance (TPM) and from the Toyota Production Systems (TPS)  5 S method uses a list of five Japanese Words: Seiri (Sort), Seiton (Straighten), Seiso (Shine), Seikketsu (Standardize), and Shitsuke (Sustain)  **What is 5 Why’s?**  The 5 Whys is an interactive question-asking technique used to explore the cause-and effect relationships underlying a problem  By asking why 5 times and answering it each time we can get the real cause of the problem. It is a brainstorming techniques  **What is Kaizen?**   * Japanese term for continuous improvement | Relentless * Philosophy involving everyone in the organization to make never ending efforts for improvement     **Kanban Challenges**   * Knowledge work industry (Green –New Brown-Existing project) * Arriving optimum WIP limit * Lesser Collaboration * No Control limit on changes * Higher degree of variation * Starvation * New Product Development * Limited Estimation / Commitment | **What are the Lean Principles & Practices**    **How will you eliminate waste in Lean?**  **What are the Value Stream Mapping Terms?**  **Value Stream Mapping** is a lean management method for analysing the current state and designing a future state for the series of events that take a product or service from its beginning through to the customer  **Lead time** is the time taken from when an issue is logged until work is completed on that issue. Lead time is what customer sees  **Cycle Time** is a measure of the time a work item takes to complete. The time a user story takes to get from the backlog to the done  **W I P (Work In Progress):**Number of work units in progress.  **Throughput**: The amount of material, data, work units that enters into a system and passes to generate output. Velocity, in Agile terms, can be the similar to this  **Little’s Law**: Cycle Time = W I P / A C R\* Where \*ACR = Average Completion Rate  **Total Cycle Time**= Value Added Time + Non Value Added Time  **Cycle Time Efficiency**: (Value Added Time / Total Cycle Time) \* 100  **What is Kanban? What are its Principles & Practices?**   * Lean approach to agile development and aim is to eliminate waste * Derived from Lean Production System (TPSJIT) * Stands for Sign Board | Billboard * Kanban is to limit WIP * This is a PULL system     **Visualize the Workflow**  Split the work into pieces, write each item on a card and put on the wall  Use named columns to illustrate where each item in the workflow  **Limit W I P (Work In Progress)**  Assign explicit limits to how many items may be in progress at each stage  Image result for kanban  **Measure the Lead Time**  (Average Time to complete on item, sometimes called “Cycle Time”) Optimize the process to make lead time as small and predictable as possible  **Kanban Positive**   * Supports dynamic requirements prioritization * No time-box value * Lead & cycle time delivery can be calculated * Visualize blockage through the cumulative flow diagram | workflow * Limit Work In Progress * Manage Workflow , Pull * Faster Throughput * No Team Limit, Allows Specialist * Well defined Roles * Improves [Kaizen] Process * Expose Bottleneck |

**Illustrate Kanban Board?**



# **Agile Planning in Handy**

|  |  |
| --- | --- |
| **What are the factors in Prioritization for agile planning?**  **Financial Value Risk**    **Cost New Knowledge** | **How would you plot Risk Value Relationship?**  Image result for Risk, Value relationship agile |
| **What is financial prioritization – Source of revenue?**  **New Revenue**  Revenue originating from implementing new features or through new business models  **Incremental Revenue**  Additional revenue eliminating from the existing customers through the enhancement of features  **Retained Revenue**  Revenue the organization would have lost otherwise, if the features were not implemented  **Operational Revenue**  Better set-up, configuration, implementation, training, cycle time reductions and cost savings coming up from these  **What are the measures to calculate financial value?**  **NPV**   * The future inflows and outflows, calculated at present point in time * Expressed as an absolute value * Using just NPV to compare the financials of two themes can be misleading   **IRR**   * How quickly will the money invested will increase in value * Expressed in % terms * Can be compared across projects * Higher the rate better it is   **Pay-back period**   * How soon will the project recover the initial investment * Does not take care of the time value of money * Does not indicate the profitability of the theme * Lesser the period better it is   **Discounted pay-back**   * Incorporates the time value of money into the calculations by discounting the future inflows and outflows based on a % rate * Lesser the payback period, better it is   **What are prioritization methods?**  The prioritization methods are:-   * MoSCoW Prioritization * 100 point method * Kano analysis * Relative weighting Model   **What is relative weighting?**   * Benefit of Presence & Penalty for absence is considered * Relative weighting (1 to 9 ) is provided by the P O’ s expert judgement * Desirability for each of the theme is calculated based on the relative weights   **What is Chartering?**   * The technique of Chartering in Agile has same goal as Develop Project Charter in PMBOK * A document issued by the Project sponsor that formally authorizes the existence of a project * Agile Charter is a Short document * Agile Project Charter will contains   + Vision   + Goals   + Measures of Success   + Trade-off Matrix   + Challenges | Roadblocks | Risks   + Definition of Done (Feature | Release | Project )   + Team Names | Availability Milestones   **What is minimum marketable feature set?**  A minimum marketable feature is the smallest possible set of functionality that, by itself, has value in the market place.  **What is Release Planning in Agile?**    Image result for agile release planning | **What is MOSCOW Prioritization?**  Primarily used in Dynamic System Development Method (DSDM)  **What is 100 Point Methods?**  Each member given 100 points  Those item with the highest number of points is ranked first  Ex: 5 Items needs to be prioritized  100 points provided to each member  Item 1 get 40 Votes  Item 2 gets 30 Votes  Item 3 gets 15 Votes  Item 4 gets 10 Votes  Item 5 gets 5 Votes  **What is Kano’s Model?**  Exciters or Delighters – Features which command extra price  Linear – More the better  Threshold – Must haves  Image result for prioritization for desirability - kanos model acp  Source: http://upload.wikimedia.org/  **What is Iteration Planning?**  **Velocity Driven**  **Adjust Priorities**  **Identify Iteration Goal**  **Select User Stories**  **Split into tasks**  **Estimate tasks**  **Have Target Velocity in Mind**  **Commitment Driven**  Each story is taken, broken into tasks and estimated.  Team commitment is arrived at before including that within the iteration  **What are the Iteration tasks?**  The iteration activities are   * Be Specific, add all tasks (meetings, UT, bug fixing etc.,) * Add tasks of non-development nature-testing, documentation * Dependencies between user stories need to be handled * Task should not typically take more than a day * If tasks appear wrong, re-plan accordingly   **How will you estimate the velocity?**  **Use Historical Values**   * From the similar teams under similar technology, domain, tool landscape and working environment   **Run an Iteration**   * Run a few sample iterations and arrive at a narrow range of projected velocity per sprint * o Typically 3-4 iterations will help to arrive at a projected value which is reasonably accurate range   **Make a forecast**   * When there is no scope for using historical values or sample sprints, make a forecast * Fill an iteration capacity with tasks derived from randomly selected stories. This can serve as indicative velocity   **How will you select the right iteration length?**  To decide the right length of iteration depends on the following factors.   * Length of release * Amount of uncertainty * Stability of priorities * Ease of getting feedback * Feeling of urgency * Iteration Overhead |

# **User Stories Estimation in Handy**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **What is User Story and its format?**  User Stories provide a Light Weight approach to managing requirements for a system. Short statement of function captured in an index card or in a tool.  **Format of User Story**  **As a <role> I can <activity> so that <business value>**  E.g., As a user, I can login and gain access to the intranet, so that I can collaborate with all the organization  **What are 3 C’s of User Story?**  1.**Card** – Written on a card  2.**Conversation** –Details captured I conversations  3**.Confirmation**– Acceptance criteria confirm that the story is done  **What are types of User Story?**  **Requirements**  Functional requirements  Non-Functional requirements  **Technical User Story**  Compliance to standards and regulatory concerns  An item of work in the Software Debt backlog that pays off a chunk of debt that is undesirable and valuable to the Product Owner”  **Spikes Stories**  A story or task aimed at answering a question or gathering information, rather than implementing product features, user stories, or requirements.  Items needing investigation towards backlog grooming  **What is INVEST in User Story?**  **“Independent”**  The User story should be self-contained, in a way that there is no inherent dependency on another user story  **“Negotiable”**  User Stories, up until they are part of an iteration, can always be changed and rewritten  **“Valuable”**  A User Story must deliver value to the end user  **“Estimable”**  You must always be able to estimate the size of a user story  **“Scalable”**  User Story should not be so big as to become impossible to plan or Task or prioritize with some level of certainty  **“Testable”**  The User Story or its related description must provide the necessary information to make test development possible  **What is Definition of Ready?**  “Never pull anything into a Sprint that is not ready, and never let anything out of the Sprint that is not done”  A User story needs to meet some criteria before it can be pulled into a Sprint   * User Story defied * User Story Acceptance Criteria defined * User Story dependencies identified * User Story sized by delivery team * Scrum team accepts Ux Artefacts * Performance criteria identified, where appropriate * Person who will accept the user story is identified * Delivery team has reviewed and accepted the user story * Team has a good idea what it will mean to Demo the user story   **What is Definition of Done?**   * List of activities for producing potential shippable product increment * DoD differs for a User Story, Sprint, Release * Differs from team to team   **What is Wire Frames?**   * Popular way of creating a quick mock-up of the product * Quick and easy way to get feedback (Low fidelity prototype) * Can be created using tools such as Visio or PowerPoint or even drawn on paper * Using these tools helps clarify what Done looks and validate the approach team   **What is Personas?**   * Highly detailed fictional characters that represent majority of users and other stakeholders who may not directly use the end product * Personas are created to identify the needs of the target user base * Supplementary to Epics and helps team understand users and their requirements and goals   **What is Product Roadmap?**   * Visual overview of a product’s releases and its main components * Communication tool which provides stakeholders quick overview of primary releases and intended functionality * Created by the Product Owner * Roadmap include large areas of product functionality, and when features will be available to Market * P O considers following while building product roadmap * Market Trajectories * Engineering Constraints * Value Prepositions | **What is User Story Estimation?**   * Human Mind is tuned towards relative estimation, as compared to on an absolute basis * Relatives Estimates are preferred for  |  |  | | --- | --- | | **Story Points** | **Ideal Days** | | * Story points have numbers, but only show the relative complexity. Like a 10 point story is 2 times complex as a 5 point story * It is an estimate of effort * Use a non-linear scale – Like, Fibonacci sequence is a good measure of scale (1, 3, 5, 8, 13, 21, 34, …) * Is an cross-functional estimate * Pure estimate of size * Does not decay * Quicker | * Amount of time needed to complete – When stripped of all unnecessary tasks * Should be a single estimate – not broken into role based time * Varies with every person | team * Amenable to change |   **What are the estimation techniques in Agile?**   * Planning Poker is the game used for agile estimation:- * Estimators choose a card and open at the same time * Repeated after discussions for typically three rounds * More on a quick reasonable estimate rather than an accurate one   Image result for planning poker  **How will you split the user stories?**   * Can be split on data boundaries * Can be split alongside operations * Horizontal components * Functional and Non Functional Components * Maintain the INVEST check * Like “Shasimi” – Fish Slice   **When to re-estimate the user stories?**  Re-estimate ONLY when the relative basis of the user stories change!  **What are Story Maps?**   * Story mapping is an effective inception tool to create a product backlog in a visually structured way * It helps in building a shared understanding, identify gaps in the backlog * Story maps provide end to end view of the system * Story mapping consists of ordering of user stories into two independent dimensions * Give a Story map so arranged the first horizontal row represents a “Walking Skeleton”, a barebones but useable version of the product. Working through successive rows fleshes out the product with additional functionality   Image result for story maps examples  **How Story Maps fit into Agile Planning?**  Image result for story maps fit into agile planning  **What is Minimum Viable Product?**  - It is a strategy for avoiding development of products which customers don’t want  - Rapidly build a minimum set of features enough to deploy the product and get quick customer feedback |

# **Agile Metrics in Handy**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **What is Agile Reporting?**   * Management driven metrics * Metrics are to measure outcome, not output * “Embracing Change” requires insight on those changes * Stakeholders informed in a meaningful manner on progress * Updated by the team, metrics generated by Scrum Master * Metrics automation is encouraged. Rally supports Automation   **What are the process metrics chart?**   * Sprint | Iteration Burn down chart * Release Burn down chart | Risk Burn down chart   **What is Information Radiator?**   * Group of artifacts to communicate status * Large & easily visible to the casual observer * Understood at a glance * Changes periodically and kept up-to-date * Big Visible Charts * Transparency | Visibility * Kanban Board | Planning board | burn down charts * Information Refrigerator (Information that cannot be communicated | released to everyone in public)   Image result for information radiator examples  **What are the Earned Value Metrics?**  **Metrics Definition Calculation**  PPC Planned % Complete Iteration # / Total Iterations  APC Actual % Complete Points Delivered / Total Points  PV Planned Value PPC \* BAC  EV Earned Value APC \* BAC  CPI Cost Per Index EV / AC> 1  SPI Schedule Per Index EV / PV> 1  SV Schedule Variance EV – PV> 0  CV Cost Variance EV – AC> 0  NPV Net Present Value NPV = ∑ Ct / (1+r) t  NPV, IRR, ROI – With higher values are good for the project  **What is Iteration burndown chart?**   * Information Radiator * X Axis Sprint Days, Y Axis Efforts can be hours or story points * Graphical representation work left to do Vs Time * Warns you early if things are not going according to plan * Useful for predicting when all the work can be completed * Ideally should burn down to zero to the end of the sprint * Sprint or Release   Image result for iteration burndown chart  **What is Iteration burnup chart?**   * A burn up chart indicates the how much work team completed against the project scope * Tracks completed work and total work in two separate lines * It communicates to the project stakeholders and clients how the additional feature requests affect the possibility of meeting the deadlines   Image result for iteration burnup chart  **Share the documentation & video link for Agile Tools?**  **Atlassian - JIRA**  <https://confluence.atlassian.com/jira063/jira-documentation-683541705.html>  <https://www.youtube.com/watch?v=xrCJv0fTyR8>  **Computer Associates - Rally**  <https://media.readthedocs.org/pdf/rally/latest/rally.pdf>  <https://www.youtube.com/watch?v=FZKQXmjZXY8>  **HP - Agile Manager (AGM)**  [https://softwaresupport.hpe.com/web/softwaresupport/document/-/facetsearch/attachment/KM01084149? fileName=hp\_man\_AGM2.1\_UserGuide\_pdf.pdf](https://softwaresupport.hpe.com/web/softwaresupport/document/-/facetsearch/attachment/KM01084149?%20fileName=hp_man_AGM2.1_UserGuide_pdf.pdf)  <https://www.youtube.com/watch?v=pX5T2KzXa-0> | **What is velocity?**   * Velocity = User story points delivered in a sprint * Measure of work Team completes in a Sprint * Planning Tool * Only stories accepted by PO qualify for velocity calculation * Varies across teams * Projected Vs Actual Velocity   **Sprints SP Committed SP Accepted Velocity**  1 20 16 16  2 24 24 24  Image result for velocity chart agile acp  **What is the difference between burnup & burndown?**   * Burndown chart shows how much work or effort remaining * Burn up chart shows how much work completed   **What do you think Velocity as Metrics?**  Nothing but the team capacity based on two factors mainly requirement ability & team capability  **Difference between burndown and burnup chart?**   * Burndown chart shows how much work or effort remaining * Burn up chart shows how much work completed   Image result for burndown   vs burn up chart  **What is Release burndown chart?**   * Sprints | iteration on the X axis against story points on the Y axis * Updated during the end of the sprint to show the team’s net progress in that sprint   Image result for release burndown chart  **What is Risk Management in Agile?**   |  |  |  |  | | --- | --- | --- | --- | | **Risk** | **Probability** | **Size of Loss** | **Exposure** | | Failure of network connectivity with partner systems | 65 % | 10 days | 6.5 | | Dependent systems are incompatible during the integration | 80% | 12 days | 9.6 | | Prototyping has inconclusive evidence for the final design | 50% | 10 days | 5 | | Lack of comprehensive data from live sites | 70% | 5 days | 3.5 | | User interfaces stability across all browsers | 30% | 10 days | 3 | |  |  |  | 27.6 |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Sprint** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | | **Exposure** | **27.6** | **20** | **21** | **15** | **10** | **5** | **0** |   Image result for risk burndown chart  **HowRisk adjusted in Product Backlog?**   * The product backlog items are assessed for applicability of identified risks * Based on the applicable risks, the backlog items are re-prioritized * Outcome is what is called as “Risk Adjusted Product Backlog” * Look out for transition indicators which signal that a risk is about to occur |

# **Agile Soft skills in Handy**

|  |  |
| --- | --- |
| **What is Negotiation?**   * Happens throughout the agile project * Should not be a “Zero-Sum” Game (With Winner and Looser) * Opportunity for each view point to be presented * Business case to be fully described * Pros & Cons of different options to be considered * Most effective when interactions between participants are positive and there is some room for give and take on each side   **What is Active Listening?**   * + Is hearing what someone is really trying to convey, rather than just the meanings of the words they are speaking   + In Agile projects we need to listen for the message, not just the string of words being spoken   **Coaching Agile team requires**   * We listen intently to people * We cannot listen when we are talking ourselves * Talk Less and Listen More * Wait for others to speak * Give other people enough time to be comfortable and speak up   General: - We are not learning while our mouths are flapping  **What is Facilitation?**  When Facilitating a meeting or session the following should be kept in mind:-  - **Goals**   * Establishing a clear goal for each meeting or session can help people get engaged right from the start * Everyone to be focused on that goal   **- Rules**   * Use of Cell phones * Starting and ending the sessions on time * Respecting the views of all participants     **- Timing**   * Duration of the session to be informed well in advance * Someone could be designated as time keeper * Determine in advance when the session breaks will take place   **- Assisting**   * Facilitator to ensure meeting is “Productive” and is focused on session goal and all members participate * Make sure junior or quieter members voice their opinions (Everyone)   **What is Conflict Resolution Techniques?**  **Collaborating | Problem Solving**  Incorporating multiple view points and insight from different perspectives. Requires open dialogue that typically leads to consensus and commitment  **Force | Direct | Competing**  Push one’s view point at the expense of others; Offering win-lose solutions  **Compromising | Reconcile**  Searching for solutions that bring some degree of satisfaction to all the parties in order to temporarily or partially resolve the conflict  **Smoothing | Accommodating**  Emphasize areas of agreement than areas of difference  Conceding one’s position to the needs of others to maintain harmony and relationship  **Withdrawal | Avoiding**  Retreat; Cool off; Postponing issues  **What are the Agile Community Values?**  The following are the agile community values are:-   * Vision * Servant Leadership * Trust * Collaboration * Honesty * Learning * Courage * Openness * Adaptability * Lead the Change * Transparency   **What are the Feedback techniques?**  The feedback techniques in agile are:-   * Prototyping * Simulation * Demonstration * Pair Programming * Unit testing * Continuous Integration * Daily Stand-up meeting   Sprint Planning | Sprint Review | Sprint Retrospective | **What is Servant Leadership?**  Agile promotes Servant Leadership Model that recognizes it is the team members, not the leader, coach, scrum master, who get the technical work done and achieve a business value  It focusses the leader on providing what the team member needs, removing the impediments to progress, and performing, supporting tasks to maximize the teams productivity  Four Primary duties are:-  **- Shield the Team from Interruptions**   * Isolate and protect team members from diversions, interruptions * Shield from request for work that are not the part of the project * Physically co-locating team members is an effective way to prevent external interference’s   **- Remove Impediments to Progress**   * Ex: Daily Stand up Meeting   **- (Re) Communicate Project Vision**   * Reiterate common vision * A developers desire for simplicity or a new technology can cause his or her work to diverge from user’s requirements * Communicating and recommunicating the project vision helps stakeholders recognize these divergences and bring them back in line with the project objectives   **- Carry Food and Water**   * Provide essential resources which the team needs * Ex: Tools, Encouragement, Compensation * Sincere “Thank You” to someone for their hard work * Celebrate victories as the project progresses (Not wait until end) * Training and other professional development activities   **What is Colocation?**  Colocation promotes face-to-face communication and fosters early query resolution and promotes osmotic communication  Osmotic communication means that information flows into the background hearing of members of the team, so that they pickup relevant information as though by osmosis. This is normally accomplished by seating them in the same room.  **What is Caves and Common?**   * Common area for collaboration – Open flow for information * Caves for Privacy * Intense Problem Solving * Create Solitude * Private Phone Calls * Research * Rocking Silently and Weeping   **What is Agile Tooling?**   * It is a collection of technology tools to bring the distributed team together * It is useful for the team to share ideas, code, knowledge, practices * Goal of Agile tooling is to increase bondage between teams when they are distributed by using technology * Agile tools – Webcams, WebEx, Chat & Shared electronic documents   **What is Participatory Decision Model?**  Participatory decision models present different ways to engage the team in the decision making progress:-  - Mechanism in place for making decisions while still keeping everyone engaged in the project. - It is not realistic to expect the team to achieve total agreement on all issues and decisions  - These mechanism are called Participatory Decision Models  **1. Simple Voting**  A simple “For” or “Against” Voting  **2. Thumps Up | Down | Sideways**  Show of Thumps up, down or sideways around the room is a more efficient way of achieving a simple vote still allowing some time to discuss other options  Members holding the thumb sideways may be neutral on the idea, might have a conflict, concern or question that needs further investigation  **3. Jim Highsmith’s Decision Spectrum**  Team members indicate how they feel about a decision by placing a checkmark on the spectrum ranging from “Fully in Favor” to “Mixed Feelings” to “Absolutely No” or Veto  It could be created on a whiteboard with permanent markers  **4. Fist of Five**  People vote by showing the number of fingers that indicates their degree of support  One Method: A first (No Fingers) means to support, while raising five fingers indicates full support  Other Method: (Most Popular)  **One Finger:** I totally support this option  **Two Fingers:** I support this option with some minor reservations that we probably don’t need to discuss  **Three Fingers:** I have concerns that we need to discuss  **Four Fingers:** I object and wants to discuss the issue  **Five Fingers:** Stop, I am against this decision |

# **Scaling Agile Framework in Handy**

|  |  |
| --- | --- |
| **What is Scaling in Agile?**  Scaling in Agile means going from a few agile teams to multiple, or even hundreds of, agile development teams.   * SAFe is applicable for enterprise level. SAFe works with 5-12 teams (50 – 125+ individuals) * Scrum is applicable for team level. Scrum team size is 7+- 2   **Definition of SAFe or What is SAFe?**  SAFe is an [agile software development](https://en.wikipedia.org/wiki/Agile_software_development) framework designed by Scaled Agile, Inc.  SAFe is a freely revealed knowledge based of integrated, proven patters for enterprise Lean-Agile development.  Visit - http://www.scaledagileframework.com/  **What are the three SAFe benefits or achievement?**  The three benefits of SAFe are:-  1. Synchronizes alignment 2. Promotes collaboration 3. Coordinates delivery for large numbers of teams  **What are the different levels of SAFe 4.0?**  There are 4 levels of Safe namely:-   * Team Level * Program Level * Value Stream Level * Program Level   **What are the dimensions in Scaling SAFe?**   * Coordination among teams * Collocation * Large Product Backlog * Product Integration * Architecture Development * Code Quality   **What are the Scaling Practices?**   * Release Planning * Scrum of Scrum | Meta Scrum * Community of Practice * Product Backlog Refinement * Joint Retrospective * Integrated Sprint Planning   **History of SAFe** | **What are the different framework available in Scaled Agile Model?**  The different Scaling Agile frameworks are SAFe, LeSS, DAD, Nexus. Out of which **SAFe is the most popular one**  **What are the SAFe Roots?**    **Agile**  Scrum - Team Roles and Ceremonies, Lets Sprint  Extreme Programming: Continuous Integration, Test First Development, Agile Architecture, Spikes, Refactoring  Kanban: Thinking on flow, demand management and Limiting work in Progress  **Lean**   * Respect People * Lean Leaders * Kaizen * Focus on Value and Value Stream   **Product Development Flow**   * Take an economic view * Actively manage the queues * Understand and exploit variability * Reduce batch sizes * Apply WIP constraints * Control flow under uncertainty: Cadence and Synchronization * Get feedback as fast as possible * Decentralized Control   **How Lean Agile budgeting differs from Traditional Cost Center budgeting?**  In Lean Agile Budgeting mainly for fund value streams, not for projects  Fund value stream provides for full control of spend, with  -No costly and delay inducing project cost variance analysis  -No resource reassignments  -No blame game for project overruns  In Cost center budgeting creates overhead and friction, lowers velocity  Both will does not match. |

|  |  |
| --- | --- |
| C:\Users\JS5027377\Desktop\safe4_0bigpicture3.jpg | |
| **Team Level ->Benefits of an agile team**   * Support Agile Teams with software, firmware, and hardware * Apply Kanban for flow * Built-In Quality covers software and hardware * XP software guidance * Empowered, self-organizing, self-managing, cross-functional team * Delivers valuable, tested, working system every two weeks * Uses a team framework which combines the best of Scrum project management, XP-inspired technical practices and Kanban for flow * **Value delivery via User Stories**   **Value Stream Level -> benefits**  For people building the world’s largest and most critical systems:   * Establish governance with an Economic Framework and Value Stream roles * Build integrated Solutions for Customers * Manage fixed and variable Solution Intent * Manage the flow of Capabilities with the Value Stream Kanban * Apply cadence and synchronization * Coordinates development of large Solutions * Synchronizes multiple ART Value Streams * Manages Solution Intent * Integrates suppliers as partners * **Value delivery via Capabilities** | **Program Level -> benefits to a team of agile teams**   * Manage the flow of work through the Program Kanban system * Apply objective Milestones to plan, manage, and measure progress * Have enhanced guidance for program execution * Have enhanced communication with new Vision and Roadmap updates * Customize to your context with Value Stream Level constructs * Self-organizing, team of agile teams * Delivers working, tested full system increments every two weeks * Operates with vision, architecture, and UX guidance * Common iteration lengths and estimating * Face-to-face planning for collaboration, alignment, and adaption * **Value delivery via features and benefits**   **Portfolio Level -> benefits**   * An Enterprise may have multiple Portfolios * Enhance Lean-Agile Budgeting with Value Stream funding, “CapEx and OpEx” * Coordinate Value Streams in a Portfolio * Organized around the flow of value * Lean-Agile budgeting empowers decision makers * Kanban system provides portfolio visibility and WIP limits * Enterprise architecture guides larger technology decisions * Objective metrics support governance and improvement * **Value delivery via Epics** |

T E A M L E V E L

|  |  |  |  |
| --- | --- | --- | --- |
| **Roles / Teams** | **Events / Ceremonies** | **Artifacts** | **Value Delivery** |
| Agile Teams  Product Owner  Scrum Master | I P Sprints  Sprint Planning  Backlog grooming  Daily Stand up  Sprint Demo  Sprint Retrospective | Team Backlog (Include NFR’s)  Team PI Objective  Sprint Goals  Working Software  Spike  Metrics  Burndown | User Stories |

P R O G R A M L E V E L

|  |  |  |  |
| --- | --- | --- | --- |
| **Roles / Teams** | **Events / Ceremonies** | **Artifacts** | **Value Delivery** |
| Product Management  Release Management  System Team  DevOps  Business Owner  System Architect  Release Train Engineer  UX Architect | P I Planning  System Demo  Inspect & Adapt Workshop  Problem Solving  SOS  P O Sync | Product Roadmap Vision  Predictability  Program Backlog  Program Objective  Team Backlog  Non Functional Requirement  Architectural Runway  Business & Architecture Feature  PI Objective Metrics | Feature |

V A L U E S T R E A M L E V E L

|  |  |  |  |
| --- | --- | --- | --- |
| **Roles / Teams** | **Events / Ceremonies** | **Artifacts** | **Value Delivery** |
| Value Stream Engineer  Solution Architect Engineer  Solution Management | Solution Demo  ART Sync  Pre & Post PI Planning  Inspect & Adapt  Predictability Measure | Value Stream Backlog  Value Stream Objective  Vision  Roadmap | Capability |

P R O G R A M P O R T F O L I O L E V E L

|  |  |  |  |
| --- | --- | --- | --- |
| **Roles / Teams** | **Events / Ceremonies** | **Artifacts** | **Value Delivery** |
| Program Portfolio Management  Epic Owner  Enterprise Architect | Budget Planning [CapEx & OpEx]  Strategic Investment Planning  Kanban Portfolio (Epic) Planning | Portfolio Backlog  Investment Themes  Business and Architecture epics  Portfolio Vision  Metrics | Epics |

|  |  |
| --- | --- |
| **SAFe 4.0 Roles**      **Role of Product Owner in SAFe**    **Role of Scrum Master in SAFe**    **III C SAFe 4.0 Principles** | **What is SAFe Foundation?**  The foundation for SAFe are Core Values, Lean Agile Mindset, and SAFe Principles & Implementing 123    **I Lean Agile Leaders**   * Lead the change * Know the way; emphasize lifelong learning * Develop people * Inspire and align with mission; minimize constraints * Decentralize decision-making * Unlock the intrinsic motivation of knowledge workers   **II Community of Practice**  An informal group of individual collaborators and other experts, acting within the context of a portfolio, that has a mission of sharing and advancing practical knowledge in one or more relevant domains  **III A SAFe Core Values**     * Built-In Quality – Quality comes first, but not in the last “*You can’t scale crappy code*” * Program Execution – Ability to build large systems incrementally, in a predictable and reliable manner * Alignment – Teams are empowered, for certain, but they work together to accomplish the larger program and enterprise mission * Transparency – Build trust via transparency, you can’t manage a secret   **III B Lean Agile Mindset**    **IV Implement SAFe 1-2-3**  The steps involved in implementing SAFe 1-2-3 process are:-  1. Train change agents 2. Train executives, managers & leaders 3. Train teams and launch train    **Train Everyone, Launch Trains** |

**SAFe Positive and Challenges**

|  |  |
| --- | --- |
| **Positive** | **Challenges** |
| * **Proven, well documented, and flexible framework; lean underpinnings** * **People-centric view on agile delivery with clear roles, artifacts, events** * **Holistic, 3-tier view of value stream including Portfolio level** * **Strong Code quality (Agile Engineering and DevOps) focus** * **Established scaling framework in Marketplace** * **Constant refinement of SAFe knowledge base** | * **Prescriptive** * **Heavy Weight** * **Certification Centric** |

**Explain SAFe 4.0 in a Nutshell?**

**Levels of SAFe:**

* **Program Portfolio management** allocates finance , budget funding, program management and governance
* **Value stream** level supports builders of large and complex solutions that require multiple ART , as well as for suppliers and applicable for multiple portfolios
* **Program level** for long lived Agile Release Train, which delivers a portion of a Value Stream
* **Team level** describes the organization, artifact, role, activities and process model for the Agile Teams who power the Agile Release Train. Development Team pulls the task from the Product Backlog in the Program Increment (PI)

**SAFe Owner Level:**

* Epic Owner is responsible for Portfolio activities
* Value Stream Engineer is responsible for value stream activities
* Release Train Engineer (Chief Scrum Master) act as a chief scrum master for the train at Program Level
* Product Owner is responsible for team level activities

**SAFe Value Delivery:**

* Value Delivery via Epics in Portfolio, Capability in Value Stream, Feature in Program and User Stories in Team level

**Innovation and Planning (IP):**

* Spikes, Hackathons & Shippy days

**Inspect & Adapt (I&A):**

* What went well?
* What didn't?
* What we can do better next time?

**PI Planning:**

* PI Planning is the cadence based event that serves as the heart-beat of the agile release. Typical size is 10 weeks

**Stretch Objectives:**

* It is not included in the commitment and more even if team has low confidence in meeting PI objective

Encourage them to move it to stretch. If an item has many unknowns, consider moving it to stretch.

**Product Vision:**

Every levels of project has its own product vision. Vision is the feature view of the solution to be developed

**Roadmap:**

Roadmap guides delivery of features overtime.

**Metrics:**

Metrics is the primary measure of working solutions.

**Milestone:**

Milestone mark for specific process point on the development time

**Cadence:**

Cadence is the rhythm of event in SAFe

**Architectural Runaway:**

Architectural Runaway is the technical base for developing business initiatives and implementing features and capabilities

**System Demo:**

System Demo occurs at end of every iteration and provide the integrator feature

**Solution Demo:**

Solution Demo provides the regular cadence for objective evaluation of the solution and customer feedback

**Solution Intent:**

Solution Intent represents the repositories for storing, managing and communicating knowledge of the solution

**Solution Context:**

Solution Context, identifies the critical aspects of target solution environment and its impact on usage, installation, operation and support of the solution

**Enabler:**

Enabler encapsulate exploration and the architectural and infrastructure development activities

**Weighted Shortest Job First (WSJF):**

WSJF is an economic model for prioritizing jobs based on the product development flow

WSJF = Cost Of Delay/Job Size

**Agile Release Train:**

Agile Release Train is a long lived team of agile team, typically consisting of 50-120 people (dun bars number)

**Books for References**

1. “Coaching Agile Teams” - by LyssaAdkins
2. “Quiet Leadership” - by DavidRock
3. The Wisdom of Team – by Jon RKatzenbach
4. Art of Agile Development - by James Shore and Shane Warden
5. “Emotional Intelligence Coaching” by Steve Neale; Lisa Spencer-Arnell; LizWilson
6. “Applied EI” (2006) by Sparrow andKnight
7. Agile Project Management with Scrum by Ken Schwaber
8. The Software Project Manager’s Bridge to Agility by Michele Slinger and Stacia Broderick
9. Agile Product Management with SCRUM by Roman Pichler
10. Succeeding with Agile Software Development using Scrum by Mike Cohn
11. Agile Estimating and Planning by Mike Cohn
12. User Stories Applied for Agile Software Development by Mike Cohn
13. Agile Retrospectives by Esther Derby and Diana Larsen
14. Agile Game Development with Scrum by Clinton Keith
15. Agile Testing by Lisa Crispin and Janet Gregory
16. Coaching Agile Teams by Lyssa Adkins
17. The Leaders Guide to Radical Management by Stephen Denning
18. What is Agile?
19. Scrum Guide – Ken Schwaber, Jeff Sutherland  Refer [Scrum Guide](http://www.scrumguides.org/)
20. A guide to the SCRUM Body of Knowledge SBOK™ Guide – Scrum Study
21. PMI-ACP Exam Prep by Mike Griffiths
22. The Five Dysfunctions of a Team by Patrick Lencioni
23. Switch: How To Change Things When Change Is Hard by John P. Kotter