Agile Projects

Prakash Badhe

prakash.badhe@vishwasoft.in

Agile Software development

Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer/end user.

Agile Project Management

- **Agile project** management is an iterative and incremental approach to delivering requirements throughout the **project** life cycle.
- At the core, **agile projects** should exhibit central values and behaviors of trust, flexibility, empowerment and collaboration.

Agile Methodologies

- Scrum.
- Kanban.
- Lean (LN)
- Dynamic System Development Model (DSDM)
- Extreme Programming (XP)
- Crystal.
- Adaptive software development (ASD)
- Agile Unified Process (AUP)

Agile Methodology Core Principles

- **Customer satisfaction**
- **Welcome the change: always.**
- **Frequent Delivery**
- **Collaborative Work and motivated individuals and Organized Teams**
- **Reflect for Effectiveness:** At regular intervals, the team reflects on how to become more effective
- **Sustainable Development**
- **Continuous Attention:** Continuous attention to technical excellence and good design enhances agility.
- **Keep It Simple:** Agile is ruthless about cutting functionality that does not lend value.

Where to apply Agile..

- Agile is the use of an adaptive lifecycle instead of a predictive one.
- Predictive lifecycles define and design the product upfront, and their goal is to follow the plan and materialize the design.
- When it’s not possible to predict the product, an adaptive lifecycle is used. There you follow a feedback loop of creating small, usable parts of the product, and use them to design and create the next subset based on the feedback.

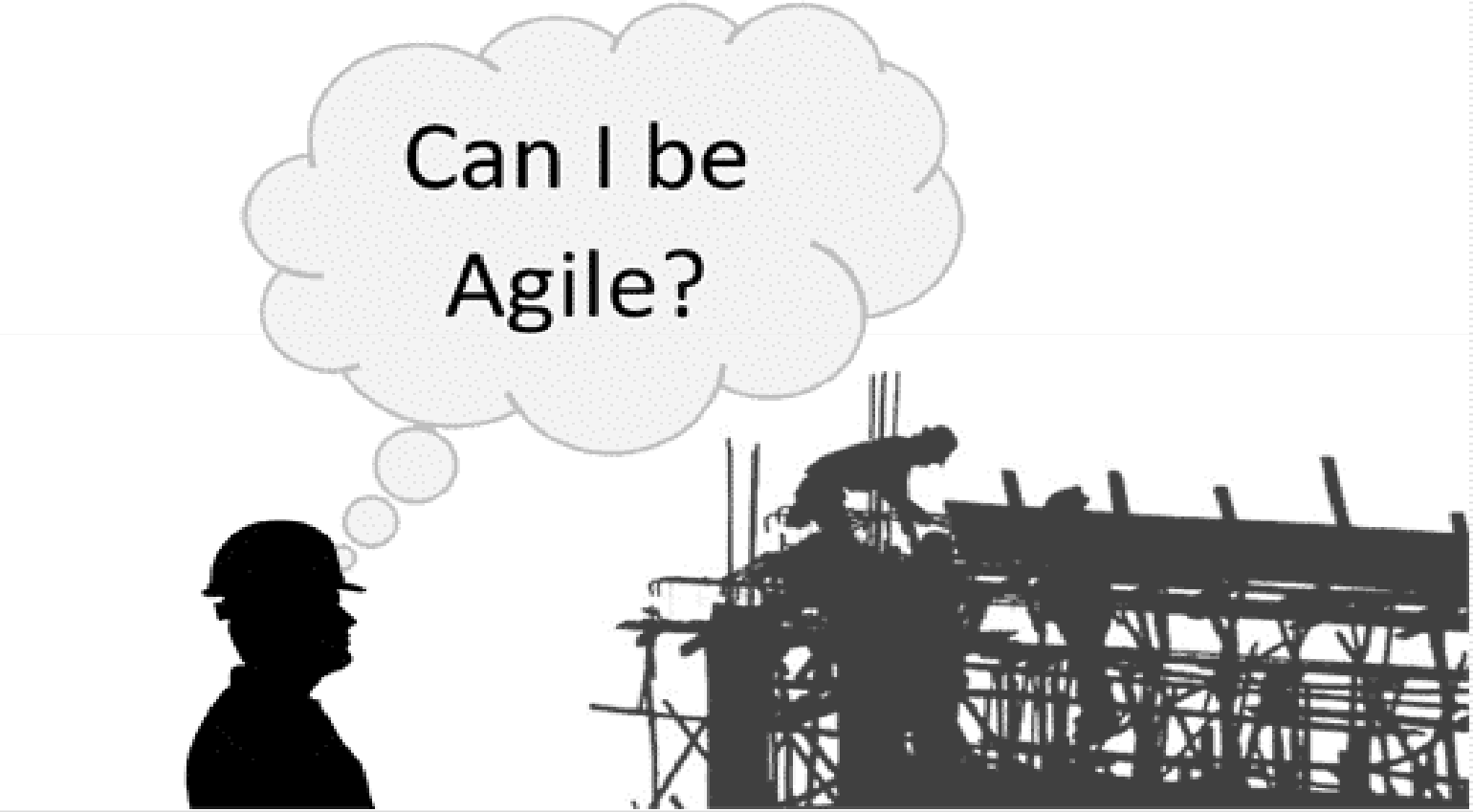
Agile for every project..?

- Agile supports two main requirements for having an adaptive lifecycle
- …both needed for enabling real feedback loops
- **Developing iteratively** : Module wise development
- **Delivering incrementally:** feature wise development and enhancement.

Agile for Software

- In Agile software process, you design, code, integrate, and test each item separately, instead of designing everything together, coding everything together, and integrating once (Waterfall).
- *Is it possible to apply agile philosophy for a building construction?*
- *Can you design a foundation without designing the rest of the building?*
- *No; there are unavoidable dependencies between the elements in this product.*

Agile for construction..?



Agile for every Project..?

- Agile cannot be used in every project, such as constructing a building.
- But you can recognize some parts of every project that have the capacity to be developed iteratively and delivered incrementally.
- For the example of a building, interior decoration is a good example of a product that can be done with an Agile system.

To be Agile or Not to be..?

- **How to Determine if a project is Fit for Agile or not ?**
- Organizations commonly adopt new processes / approaches that have proven to reach markets faster, realize quicker returns and minimize costs.
- Agile is a set of values and frameworks geared to shift efforts toward efficiency and continuous value delivery.
- Agile's success in delivering more within the challenges of the triple constraints (**scope, schedule and budget**) has gained it popularity and wider adoption in the project management world.
- Depending on the scenario, plan-driven, traditional project management or "waterfall" can provide a higher likelihood for projects' success.

How to decide on agile…?

- Not all Agile projects succeed and not all “waterfall” projects fail.
- The factors to consider when selecting the approach that best meets the needs of the organization?
- **The Nature of a Project / Product :** consider the three important aspects of a project: Scope, schedule and cost.
- **The Project / Product Environment:** the project / product delivery environment and life-cycle.

Yes for Agile..

- Idea for a new complex product and specific requirements will emerge through small experiments based on market/user needs.I.e starting a new product line or new marketing campaign.
- The product needs to meet an aggressive time line , even with limited features(For example: showing available product samples at expo or a bazaar)
- The requirements are dynamic.

As flows the water..

- The project has a checklist of rigid pre-defined requirements with predefined stages and must be completed in predictable sequence.e.g. porting a e-commerce product web site from one platform to other.
- The list of features must be completed to deliver the final product.e.g hardware installation, installing a product on network of multiple virtual machines. Like an OS image.
- The project is to be delivered within a strict set dead line.

Agile will go with customer

- Are the customers/end users available to engage throughout the development or need to be available to share the feedback.(Insurance or banking project, GST portal for Govt. of India)
- Cross functional teams are fully committed to reach quick decisions and complete the project.(Team of multi-skilled engineers working on Robot design)
- Frequent delivery of usable capabilities are required from the vendor/contractor to accommodate customer feedback(ERP Implementation)

Implement in Sequence

- The users have provided predefined requirements steps and are interested in seeing the final product only..in short duration.e.g.developing a mobile app.
- Building a new version of existing product.
- Data migration projects
- Fixed cost projects
- Testing projects.
- Projects with Limited scope.

Going by Rules

- Organizations generally identify the most fitting approach that helps accelerate delivery, minimize risk, save costs, and ultimately present a higher likelihood of success by carefully analyzing the nature of a product or project .
- With a good strategy that creates the enabling environment is always helpful.

Agile Adoption

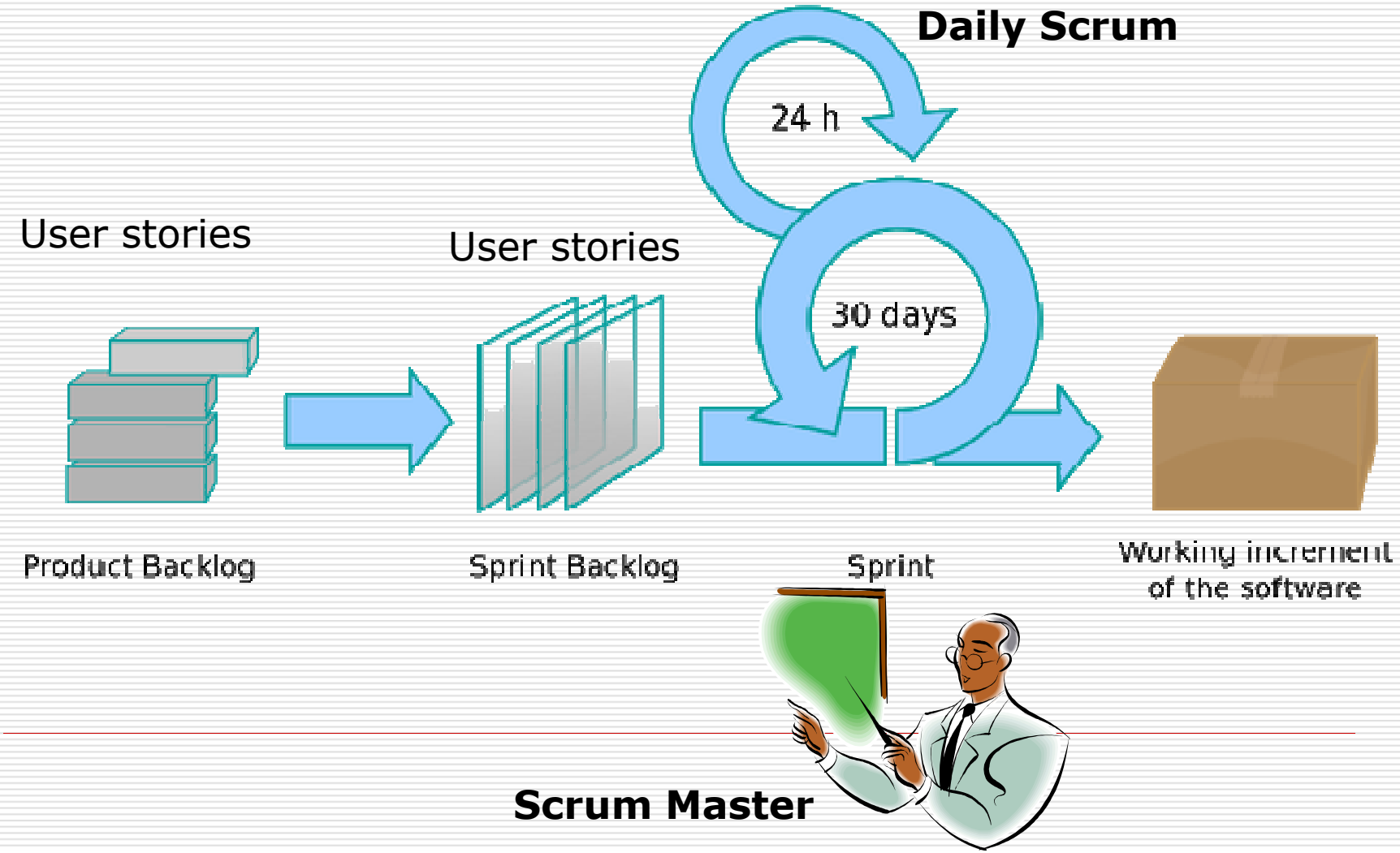
- The decision is to transition to an Agile approach, similar to any new process or change initiative, Agile adoption requires a supportive environment that spans beyond a given project / product to an *organizational transformation.*
- An Agile implementation must be supported by change management activities that foster buy-in and ensure the Agile vision and goals are clearly communicated across the enterprise.

How to start with Agile..?

- Select the methodology practices
- Form small teams.
- Assign goals and responsibilities.
- Motivate the team members, respect every individual
- Spread the word of being Agile across the organization.
- Start the work and track the progress.
- Identify scope for automation and apply it most.
- Monitor challenges and issues.
- Identify the scope for improvement and improve it.

Scrum Practice

- Scope of requirements
- Roles : Scrum Master and Team
- Client involvement and interactions
- Evolution of user stories
- Daily standup meetings
- Identify Backlogs
- Delivery schedules and iterations
- The progress monitoring and Definition of DONE.
- Feedback loops
- Burn down charts


Daily Scrum
24 h
User stories
User stories
30 days
Product Backlog
Sprint Backlog
Sprint
Working increment
of the software
Scrum Master

User story is for end-user

- A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation.
- Stories use non-technical language to provide context for the development team and their efforts.
- After reading a user story, the team knows why they are building what they're building and what value it creates.

User story

- **User stories** are one of the primary development artifacts for **Scrum** and Extreme Programming (XP) project teams.
- The User story is defined to capture a description of a software feature from an end **user** perspective.
- A **user story** is a very high-level definition of a requirement, containing just enough information so that the developers can produce a reasonable estimate of the effort to implement it.

User story considerations

- **Stakeholders write user stories.** The project stakeholders write the user stories, not the developers.
- User stories are simple enough that people can learn to write them in a few minutes, so it makes sense that the domain experts (the stakeholders) write them.
- **Use the simplest tool**. User stories are often written on index cards.
- Index cards are very easy to work with and are therefore an inclusive modeling technique.
- User stories are development tasks often expressed as “person + need + purpose.”

Use stories..

- **Scope for non-functional requirements**. Stories can be used to describe a wide variety of requirements types.
- **Indicate the estimated size**. Include an estimate for the effort to implement the user story.
- One way to estimate is to assign user story points to each card, a relative indication of how long it will take a pair of programmers to implement the story.

User stories considerations

- **Indicate the priority of features :**The requirements, including defects identified as part of the testing activities or by your operations and support efforts, are prioritized by the project stakeholders (or representatives thereof such as product owners) and added to the stack in the appropriate place.
- **Optionally include a unique identifier**. The card also includes a unique identifier for the user story, to maintain some sort of traceability between the user story and other artifacts.

Stories in Scrum and Kanban

- The User Stories fit neatly into **agile** frameworks like **scrum** and kanban.
- In **scrum**, **user stories** are added to sprints and “burned down” over the duration of the **sprint**.
- Kanban teams pull **user stories** into their backlog and run them through their workflow.

Epics

- Epics are large user stories, typically ones which are too big to implement in a single iteration and therefore they need to be disaggregated into smaller user stories at some point.
- Epics are typically lower priority user stories because once the epic works its way towards the top of the work item stack.
- It is reorganized into smaller ones. It doesn't make sense to disaggregate a low-priority epic because you'd be investing time on something which you may never get to addressing, unless a portion of the epic is high priority and needs to be teased out.

Themes

- A theme is a collection of related user stories. For example, for a university registration system there might be themes around students, course management, transcript generation, grade administration, financial processing.
- Themes are often used to organize stories into releases or to organize them so that various sub-teams can work on them.

JIRA for Agile Team

- JIRA is for every member of the software team to plan, track, and release software.
- Create user stories and issues, plan sprints, and distribute tasks across the software team.
- Prioritize and discuss the team's work in full context with complete visibility.
- Improve team performance based on real-time, visual data that the team can put to use from Jira tool.

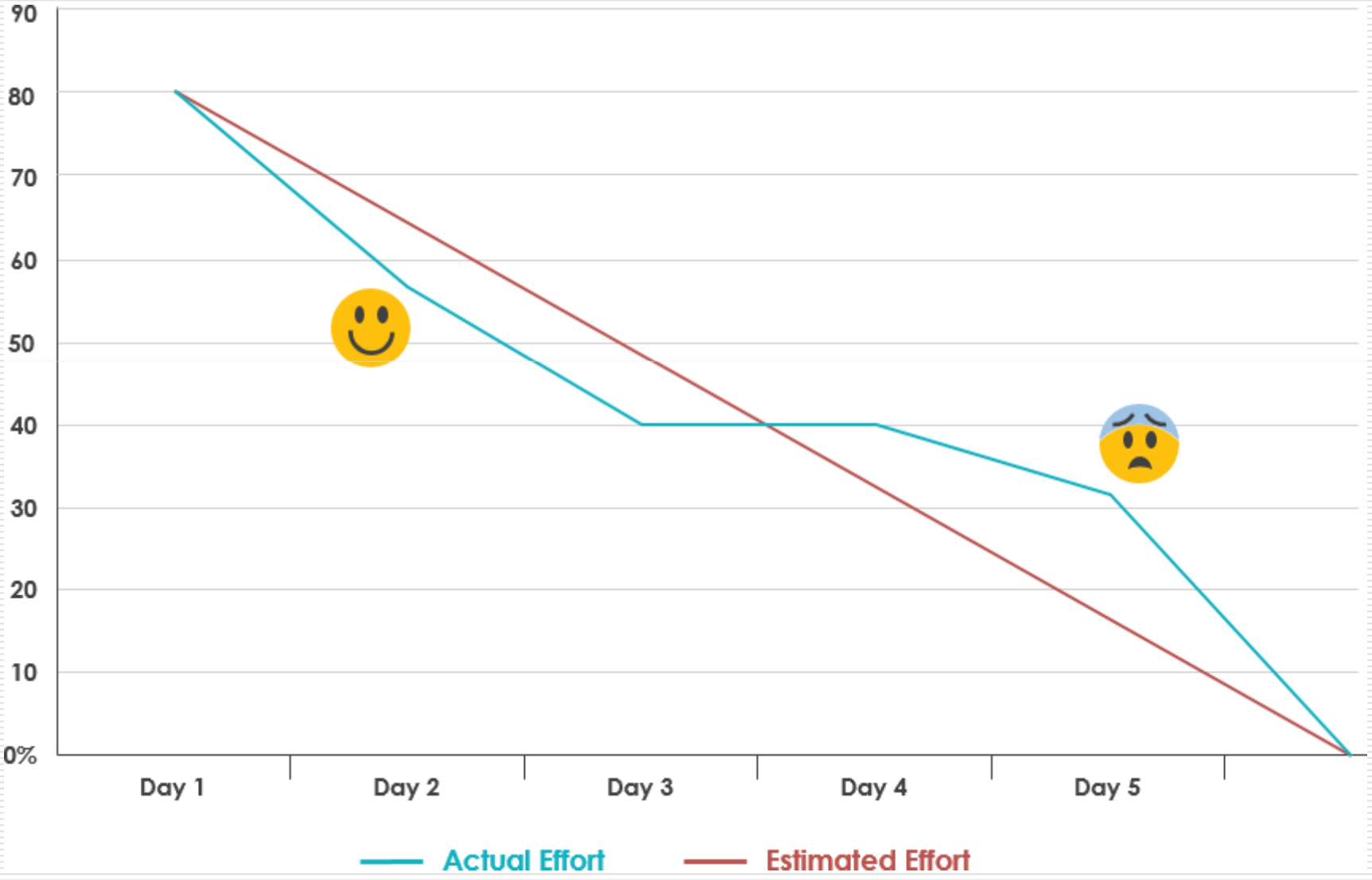
Standup meetings

- *In agile practices like scrum, kanban daily team meetings are suggested to make commitments to team members.*
- *The daily commitments allow participants to know about potential challenges as well as to coordinate efforts to resolve difficult and/or time-consuming issues.*
- A stand-up meeting is a meeting in which attendees typically participate while standing.
- The discomfort of standing for long periods is intended to keep the meetings *short.*

Burn down chart

- A **burn down chart** is a graphical representation of work left to do versus time.
- The **burndown chart** shows the total effort against the amount of work for each iteration.
- **It** is often used in **agile** software development methodologies such as **Scrum**.
- The **burn down charts** can be applied to any project containing measurable progress over time.

Example Burn Down Chart



Thanks to you!!