

The image features a horizontal banner. On the left, a dark blue circle partially overlaps an orange circle. The Protegra logo is positioned on the blue circle, with the word 'Protegra' in white and orange, and the tagline 'Business. Technology. Solutions.' below it. To the right of the circles is a photograph of a man in a black wetsuit holding a surfboard against a clear blue sky.

Protegra

Business. Technology. Solutions.

Be better

## Lean LifeCycle

# LifeCycle Principles



- **LifeCycle Mission Statement:**

The Lifecycle mission is “to provide Project Teams with the guidance needed to consistently deliver value to clients’ businesses through the creation and evolution of high quality software-enabled systems that meet clients’ needs and expectations”.

# Discussion



- **Coffee and 15 minute group discussion**
  - What was the best project team you were part of?
    - What was the leadership like?
    - What was the teamwork like?
    - How did the team innovate?

# LifeCycle Principles



- **Client Focus**
  - The Lifecycle approach reflects the active manner in which Protegra integrates the client into the entire working process to create solutions with impact. The methodology is client centric: it begins with clearly identifying the project's stakeholders and understanding their needs; it ends with a validation that the solution delivered in fact meets the documented needs without critical defects.
- **Quality and Costs**
  - To achieve quality, one must plan for it. Lifecycle incorporates significant elements of quality planning and continuous improvement to deliver quality solutions while maintaining appropriate control of the costs involved (on time and on budget), and without adding unnecessary time or expenses to the project.
- **Rigour vs. Innovation**
  - The balance between rigour and innovation is a critical component of the Lifecycle methodology. Achieving this balance is key in optimizing the quality versus cost equation discussed in the previous principle. An unnecessary level of rigour adds rigidity, time and costs to the project; too little rigour often jeopardizes the quality of the project and its solution.

# LifeCycle Principles



- **Continuous Improvement**

- Lifecycle provides guidance for projects to set objectives and define performance expectations and several opportunities for improvement, based on observing performance and comparing it to planned objectives.

- **Managing Risk**

- Risk, defined as any event whose occurrence could jeopardize project success, is a core concept that must be understood and managed effectively to be successful in managing system development projects. A risk has characteristics such as likelihood (probability of occurrence), impact (effect of risk on project), and associated cost.

- **Managing Change**

- Change is a fact of life, and for the most part necessary. Uncontrolled change is likely the most severe risk to a project – it is often referred to as *scope creep*. The scope of a project is defined by its triple-constraint of *what, when, and how much*. Change management is not only an activity that spans the full life cycle of the project; it is a mindset.

# LifeCycle Principles



- **Value based**
  - LifeCycle is a value based methodology. A critical eye should be used to ensure everything we are doing returns customer value. We should not create certain documents or follow certain processes if they add no value to our customer. The value would be realized in clear client satisfaction or our belief that they add long term value to the software development methodology and are required. (For Example, the standards of frequent builds and client interaction) LifeCycle provides guidelines to highlight what may provide value, but the project teams are tasked with the final decision on what provides value.
- **Deliverable-based**
  - Lifecycle is deliverable-based, not task-based. This enables Project Teams to plan against deliverables, and associate costs against those deliverables. In this way, costs are not associated to activities, and the exact process of completing a deliverable is seldom mandated, but left to the innovation of the Project Team to complete.

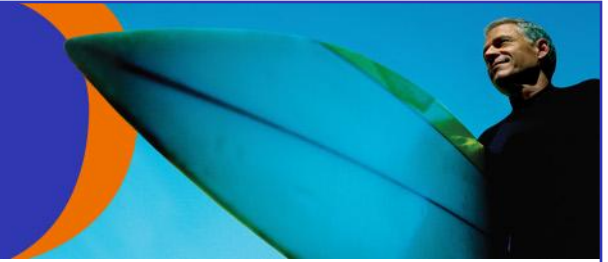
# LifeCycle Principles



- **Process Model**

- A *Process Model* is a generic statement of a project process. A model will typically define stages, flow between stages, and activities and artifacts within a stage. Process Models are broken into two categories *specification-based* and *evolutionary* [Sommerville, 1996]. Lifecycle recommends that Project Team's choose between the Incremental Model and the Spiral Model, depending on the problem the project is trying to solve.

# LifeCycle Principles



- **High Performing Teams**

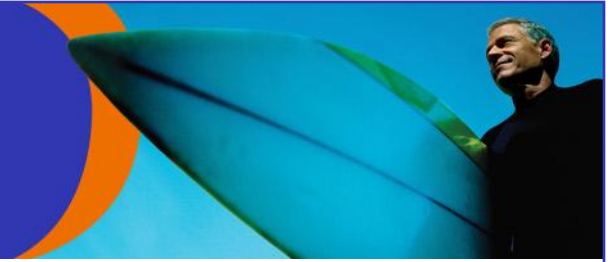
- Commitment – to the purpose and values of Protegra. Team members understand how their work fits into Protegra’s strategy and objectives; team’s efforts are aligned.
- Contribution – every team member contributes to the success of the team, this is best achieved through inclusion, confidence, and empowerment. Confidence can be bolstered by feedback, coaching, peer reviews and professional development opportunities.
- (Open, effective) Communication – only happens in an atmosphere where team members show concern, trust one another, and focus on solutions, not problems. Effective listening, giving/receiving feedback, etc.
- Collaboration – an environment that creates the most desirable outcomes for all involved parties. Achieved through team problem-solving, innovation, new approaches, etc. Follow through on commitments, high standards of performance, creativity, timeliness.
- Conflict management – how members deal with conflict, respond to it, etc.
- Change management – teams must foster changes in their methods and Protegra’s methods toward improvement.
- Connections – to the client organization, to team members, to the other project teams, to Protegra.

# Lean LifeCycle



- **Lean LifeCycle is Protegra's Lean Software Development Methodology**
- **It is a combination and solution that takes the components that add value from the following methodologies:**
  - Waterfall
  - Spiral
  - Incremental
  - Scrum
  - Agile

# Lean Software Development



- **Lean Software development is a style of software development that emphasizes customer satisfaction through continuous delivery of functional software. In contrast to traditional software development methods, lean developers liaise continuously with business clients.**
- **Their objective is to deliver working software as frequently as every two weeks during a project, and welcome changes to the requirements in response to evolving business needs.**
- **The most crucial aspect of Lean LifeCycle is the execution of the project in iterations and quick feedback loops possible because of these iterations. It is essential to note that these iterations to not just apply to construction, they also apply to the following tasks:**
  - Project Management and Planning
  - Analysis
  - Technical Design
  - Testing
  - Deployment

# Lean Software Development

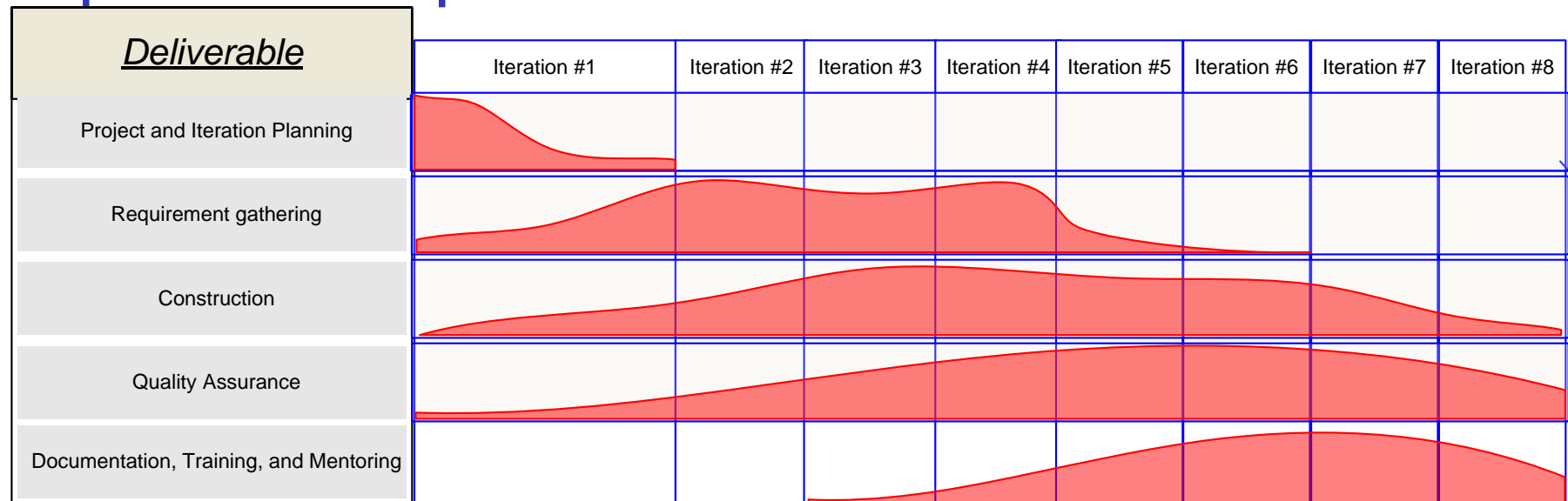


- **Iteration planning is ‘the’ key planning initiative**
  - Iterations need to be planned in conjunction with the client to accomplish the following:
    - Deliver functionality to define the cadence and tempo of the project
    - Deliver functionality to deliver real value to the client
    - Deliver functionality to reduce and minimize risk for the entire project
    - Lessons learned from one iteration must feed into subsequent iterations so that we don’t execute the project in iterations with similar results, but that we execute the project in iterations with better results.
      - We execute better, smarter, and quicker

# Lean Software Development



- Most deliverables are enhanced in every iteration (except for project plan)
- Every iteration uncovers new opportunities for improvement and learning
- Opportunities for improvement are evaluated at the end of each phase and incorporated into the tool and documentation



# Lean Software Development Principles



- **Eliminate Waste**
  - The three biggest wastes in software development are:
    - ***Extra Features***
    - ***Churn***
    - ***Crossing Boundaries***
- **Build Quality In**
  - If you routinely find defects in your verification process, your process is defective.
  - ***Mistake-Proof Code with Test-Driven Development***
  - ***Stop Building Legacy Code***
  - ***The Big Bang is Obsolete***

# Lean Software Development Principles



- **Create Knowledge**

- Planning is useful. Learning is essential.
- ***Use the Scientific Method***
- ***Standards Exist to be Challenged and Improved***
- ***Predictable Performance is Driven by Feedback***

- **Defer Commitment**

- Abolish the idea that it is a good idea to start development with a complete specification.
- ***Break Dependencies***
- ***Maintain Options***
- ***Schedule Irreversible Decisions at the Last Responsible Moment***

# Lean Software Development Principles



- **Deliver Fast**
  - Lists and queues are buffers between organizations that simply slow things down.
  - ***Rapid Delivery, High Quality, and Low Cost are Fully Compatible***
  - ***Queuing Theory Applies to Development, not Just Servers***
  - ***Limit Work to Capacity***
- **Respect People**
  - Engaged, thinking people provide the most sustainable competitive advantage.
  - ***Teams Thrive on Pride, Commitment, Trust, and Applause***
  - ***Provide Effective Leadership***
  - ***Respect Partners***
- **Optimize the Whole**
  - Brilliant products emerge from a unique combination of opportunity and technology.
  - ***Focus on the Entire Value Stream***
  - ***Deliver a Complete Product***
  - ***Measure UP***

# Lean LifeCycle Mandatories



- **Lean LifeCycle Mandatory Requirements**
  - **Maximum Iteration length of 1 month**
    - Day 1 planning
    - No changes after Day 1
    - Day 30 – Demo and installation
- **Vision must be created before iterations start**
  - Technical (proof of concepts, architecture and vision)
  - Business (storyboard, requirements document at least 20% complete)
  - Delay decisions as long as possible. But make architectural or other decisions early if they are required.
- **Daily progress meetings (15 minutes maximum)**
  - 3 questions
  - What have you worked on since last meeting
  - What are you planning to work on next?
  - Do you have any issues you require assistance on?

# Lean LifeCycle Mandatories



- **Test driven development**
  - Function specification/test case combined if it makes sense.
  - Developer also writes the test cases.
  - Test driven development where it makes sense. This approach may apply more to client functionality than batch or nightly processes.
- **Requires full access to business users or stakeholders.  
(Preferably on site)**
- **Refocus sign off process on how not what**
  - Typically the sign off process has focused on what we are building. This is an example of point based solution development. As we move into Lean Development it is more important to focus on how we are executing the project or set based development. The details will be defined when appropriate if we define and agree how we will execute. Subsequent sign offs on what we are developing are still required, but should be routine if we have signed off how we are executing.

# Lean LifeCycle Mandatories



- **Frequent Builds**
  - Build at least weekly internally
  - Build at least monthly for customer deployment
  - It is acceptable that the first deployment to the client may not occur until Iteration2.
- **Sub-deliverables/task size about 1 week maximum (if possible)**
- **Visual Project Management**
  - This is done with a combination of providing the project plan and tasks on SharePoint or other visual means

# Lean LifeCycle Implementation Guidelines



- **Refocus requirements discussion on how not what**
  - Waterfall (Point based)
    - Document the scope or the what before we start
    - Use the scope document to manage change control
    - Discussions on new and modified scope can be difficult as these requests usually occur later in the project and require additional budget and schedule. Changes are avoided for these reasons.
    - Discussions become adversarial
  - Lean (Set based)
    - Document the process on how we will define the scope and make decisions. Define at a high level what you need and how you will refine it.
    - Use the process agreement to manage change control
    - Joint Collaborative discussions on new and modified scope and how we can manage them in the project. Because we deliver in iterations, changes can be specified earlier and can often be accommodated without additional budget and schedule. Changes are welcomed.

# Lean LifeCycle Implementation Guidelines



- **Project team will create project plan**
- **Project team will define iteration content and length (in conjunction with the client)**
- **Project team will recommend what aspects of lean development they recommend using.**
  - What optional components
  - To what level the mandatory components will be used.
- **Delivery Management will review with the team the plan and track how lean development is planned to be and how it is working**
- **We will review, refine, and re-plan for future projects/iterations based on lessons learned.**

# Lean LifeCycle Rhythm



- **There are four rhythm's that are critical to create and execute on a Lean LifeCycle project. These rhythms focus on the monitoring and oversight of the project performance to ensure the project is executing according to the project plan. These four rhythm's are**
  - Daily
  - Weekly
  - Monthly
  - Iteration

# Daily Rhythm



- **The following tasks or activities are mandatory to be done every day on the project. The time of the day and owner of these tasks can be customized by the team. These activities are:**
  - Daily Stand-up Meetings
  - Daily time entry
  - Entry of New Issues and modification of existing Issues
  - Entry of New Risks and modification of existing Risks
  - Daily Review of Issues and execution of required actions

# Weekly Rhythm



- **The following tasks or activities are mandatory to be done once a week on the project. The day of the week and owner of these tasks can be customized by the team. These activities are:**
  - Red/Green/Yellow report of Project Status
  - Updating of Project budget and cost and creation of Client Status Report
  - Delivery Management Weekly Review of status report and issue list
  - Client Relationship Management Weekly Review of status report and issue list
  - Weekly Review of Status Report, Issues, and Risks with the Client

# Monthly Rhythm



- **Invoice Creation**
- **Next month Forecast**

# Iteration Rhythm



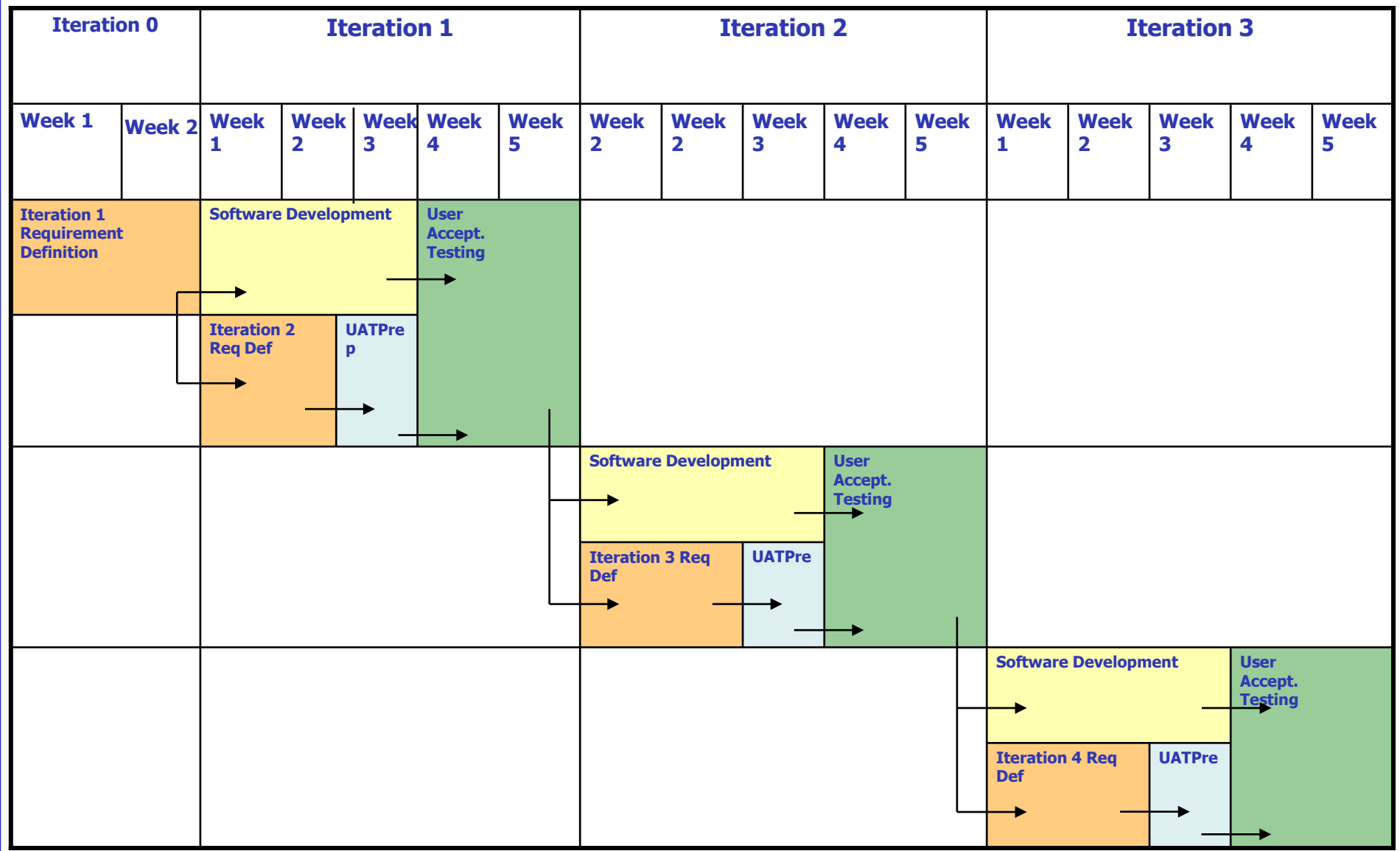
- **The following tasks or activities are mandatory to be done once an iteration on the project. The day of the iteration and owner of these tasks can be customized by the team. These activities are:**
  - **Validate Iteration Scope**
    - This section is comprised of the following activities:
      - Incorporate previous iteration feedback
      - Validate Iteration plan due to confirmed scope
    - Schedule: First couple of days of the iteration
  - **Review and Mitigate Risk**
    - This section is comprised of the following activities:
      - Hold Risk review meeting
      - Execute Risk Mitigation strategies
    - Schedule: First couple of days of the iteration
  - **Build Software Components**
    - Build the agreed to Software Components
    - Schedule: First three weeks of the iteration

# Iteration Rhythm



- **Certify Software Components**
  - This section is comprised of the following activities:
    - Test Planning
    - System and Integration Testing
    - User Acceptance Testing
  - Schedule: Last two weeks of the iteration
- **Create Component Blueprints/Use Cases for Next Iteration**
  - Create the Blueprints and Use Cases for the next iteration
  - Schedule: Anytime during the iteration
- **Iteration Closeout**
  - This section is comprised of the following activities:
    - Hold iteration closeout meeting
    - Document iteration closeout meeting and assign actions
  - Schedule: Immediately after the iteration

# Iteration Rhythm

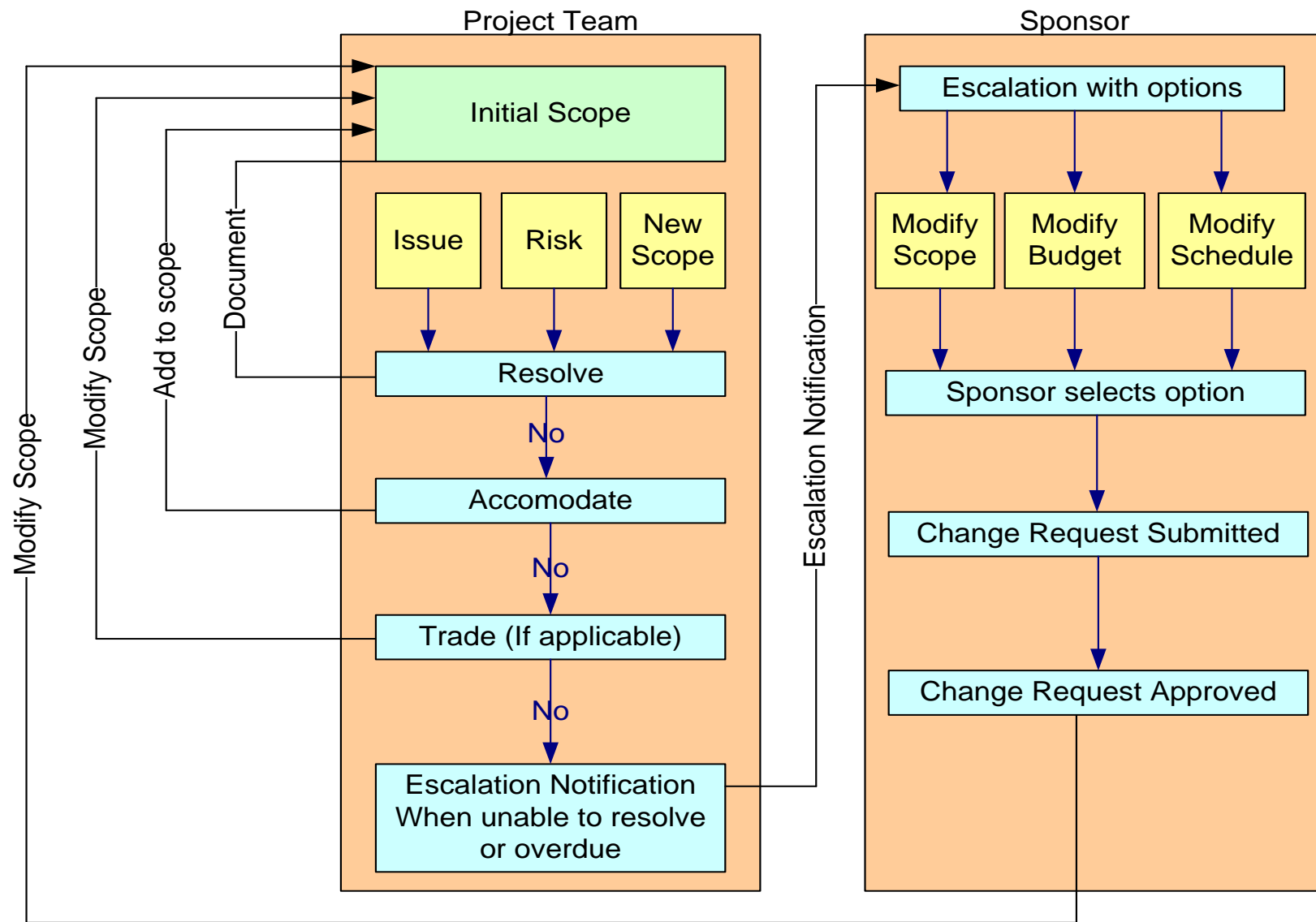


# Lean LifeCycle Percussion



- **In addition to the Rhythm of a Lean LifeCycle project, the percussion of the project is of critical importance. As in the music metaphor, while the Rhythm defines the melody or music, it is the underlying percussion that allows the rhythm to be paced, pleasing, and successful. In Lean LifeCycle, the percussion elements are the basics of project management. There are five types of Lean LifeCycle Percussion:**
  - Internal Team Percussion
  - External Team Percussion
  - Risk Management Percussion
  - Issue Management Percussion
  - Change Management Percussion

# Lean LifeCycle Percussion



## Internal Team Percussion



- **The following procedures will be followed when to define the internal team discussion. Once we have defined the scope and have held the project kick offs, the following procedures will be used to address Issues, Risk, and New Scope Items.**
  - **Resolve:** Attempt to resolve the issue, risk, or scope internally. This could be a situation of communication or interpretation
  - **Accommodate:** Attempt to accommodate the issue, risk, or scope within the given project scope.
  - **Trade:** Attempt to accommodate the issue, risk, or scope by delaying or deferring scope items.
  - **Escalation Notification:** If the tactics of Resolve, Accommodate, and Trade are not successful, the Client project manager will be contacted via email to notify that an escalation form will be submitted. The escalation form will then follow within 24 hours. These escalation items will also be tracked on the status report.

## External Team Percussion



- **The following procedures will be followed once we have escalated to the sponsor.**
  - Escalation with options: Present escalation with options for a client decision.
  - Sponsor Selects Option: Sponsor will decide to change budget, scope, or schedule
  - Change Request Submitted: Change request will be created to modify initial scope.
  - Change Request Approved: Change request will be approved by both parties.

# Risk Management Percussion



- ***Risk:*** An item that might occur related to the successful completion of a project. A project risk is tracked, publically available to the team, together with any mitigation strategies that emerge. It usually has a ranking associated with it to indicate its severity or urgency, and it also is subject to escalation if the mitigation strategy cannot be affected. Once a risk occurs, it becomes an issue, requiring resolution.
- **The following activities will be done as part of the risk management percussion:**
  - Initial Risk meeting with Team
  - Initial Risk meeting with Client
  - Iteration closeout meeting risk review
- **In addition risks will be reviewed on a weekly basis with the status report highlighting which risks have changed.**

# Issue Management Percussion



- **Issue:** An item related to the successful completion of the project that requires information, discussion, or a decision to resolve. The following activities will be done as part of the issue management percussion:
  - Project 'waits' are entered as issues if the team can not solve internally in 4 hours AND impact is either currently being experienced on the project or will be experienced on the project if not solved in 48 hours.
  - These issues will be categorized as yellow if some progress can still be made on this or other tasks. These issues will be categorized as red if progress is halted.
  - These issues will be entered with a due date
  - Unresolved Issues will be escalated to client PM at the end of each day. Date of escalation will be recorded.
  - Overdue issues will be escalated to project sponsor at the end of each day. Date of escalation will be recorded.
  - In addition issues will be reviewed on a weekly basis with the status report highlighting which issues are overdue.

# Change Management Percussion



- **The following activities will be done as part of the change management percussion:**
  - Change Management item review
  - Iteration closeout change management review
- **In addition change requests will be reviewed on a weekly basis.**

# Break



- **Coffee and 15 minute group discussion**
  - What was the worst project team you were part of?
    - What was the communication like?
    - How did the team respect one another?
    - How did the team struggle?

# LifeCycle Project Stages



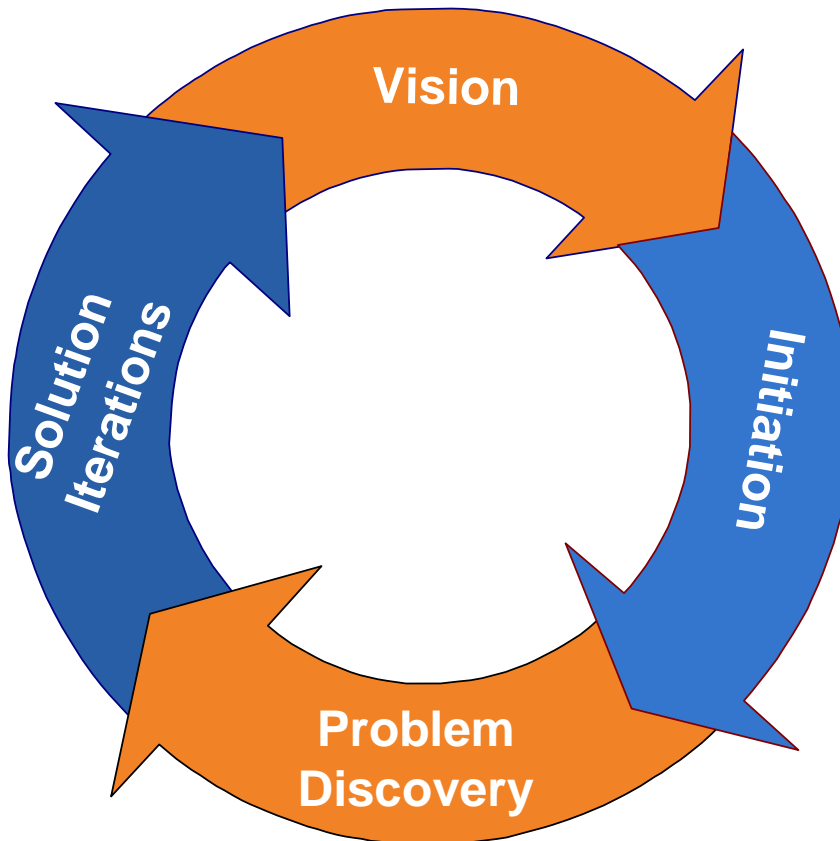
- **LifeCycle segments its project phases into three stages. These stages are:**
- **Pre-Iteration Project Stage**
  - Vision Phase
  - Initiation Phase
  - Problem Discovery Phase
- **Solution Iteration Project Stage**
  - Planning Phase
  - Solution Discovery Phase
  - Construction Phase
  - Certification Phase
  - Deployment Phase
- **Post Iteration Project Stage**
  - Project Closeout Phase

# Pre-Iteration Project Stage



## Vision Key Client Deliverables

- Statement of work
- Contract



## Initiation Key Client Deliverables

- Client Project Charter
- Iteration Plan and Charter
- Development Team Charter
- Risk Register
- Status Report
- Issues/Decisions/Defects List
- Project Plan

## Problem Discovery Key Client Deliverables

- Software Requirements
- Architecture Blueprint
- Traceability Matrix

# Pre-Iteration Project Stage



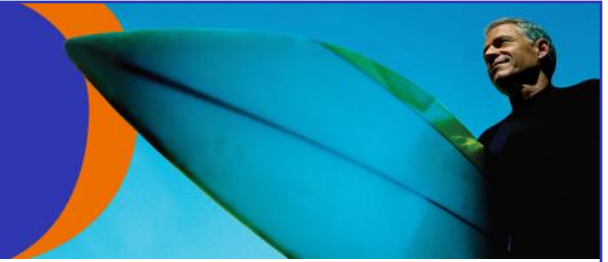
- **Vision Phase**

- In this phase, the project is conceived and initially scoped through the development (or a review) of a vision for the system. This phase will define the initial triple-constraint of *what*, *when*, and *how much*, which will provide the foundation for the Business and Technology Model.

- **Initiation Phase**

- During this phase the Project Team will be assembled and will be given the responsibility of delivering the vision for the project. The Team Charter deliverable will define project roles, responsibilities, and rules of engagement for the remainder of the project.
- A key component of this phase is the deliverable of creating an executable plan.

# Pre-Iteration Project Stage



- **Problem Discovery Phase**

- Problem Discovery describes the process of interviewing all project stakeholders and identifying their needs, and their needs' importance. Techniques such as Business Process Analysis, Value Stream Mapping, and conceptual prototyping may be used to elicit stakeholder requirements. Once the requirements are gathered and documented, the Project Acceptor will sign-off on the requirements on behalf of the stakeholders, which will effectively baseline the feature scope of the project. The start of the Requirements traceability processes will initiated during this phase. In addition to requirements gathering, risks will be identified and risk management plans will commence.
- A critical deliverable of this phase will be the iteration charter and plan for the project which proposes the feature set to be delivered per iteration.
- A key component of this phase will be Iteration 0. Iteration 0 typically is comprised of the following:
  - Initial Architecture design decisions and investigation
  - Environment set up
  - Orientation
  - The Testing Planning for the project also traditionally starts here.

# Solution Iteration Project Stage



- **Planning Phase**
  - The planning phase is a short phase to confirm the requirements and execution plan of the iteration.
- **Solution Discovery Phase**
  - The Solution Discovery Phase builds upon the general business requirements of the Problem Discovery phase and defines them in greater depth. Initial domain modelling will occur in this phase to aid the Application Architect's understanding of the overall domain problem, and will help in extracting further requirements.
- **Construction Phase**
  - Activities within this phase largely depend upon the type of system under construction. Enterprise vision, system acquisition, supply and software development projects have different deliverable requirements. Lifecycle's current focus is on the software development deliverables, however the fundamental Lifecycle principles apply to all solution types.

# Solution Iteration Project Stage



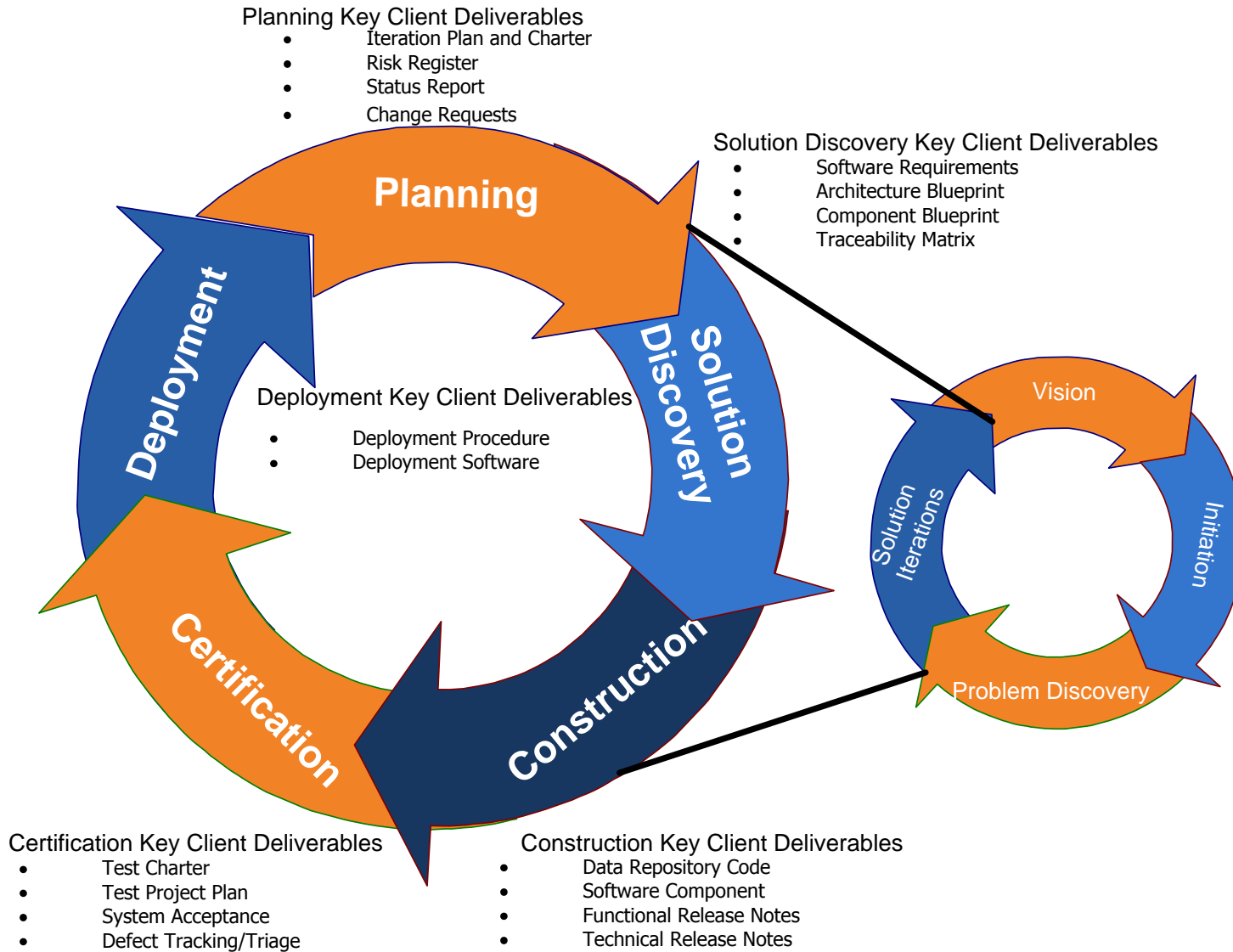
- **Certification Phase**

- System Testing (including Functional, Integration and/or Interface) and User Acceptance Testing will be performed in this phase. The Project Team will be supporting the Quality Assurance Team throughout the course of this phase. Quality reviews will be performed to validate when the delivery goes to production.

- **Delivery Phase**

- This phase develops deliverables that will support the system components in production. The Deployment Procedure, Deployment Software, and User Documentation will be produced during this phase.

# Solution Iteration Project Stage



# Post Iteration Project Stage



- **Project Close Phase**

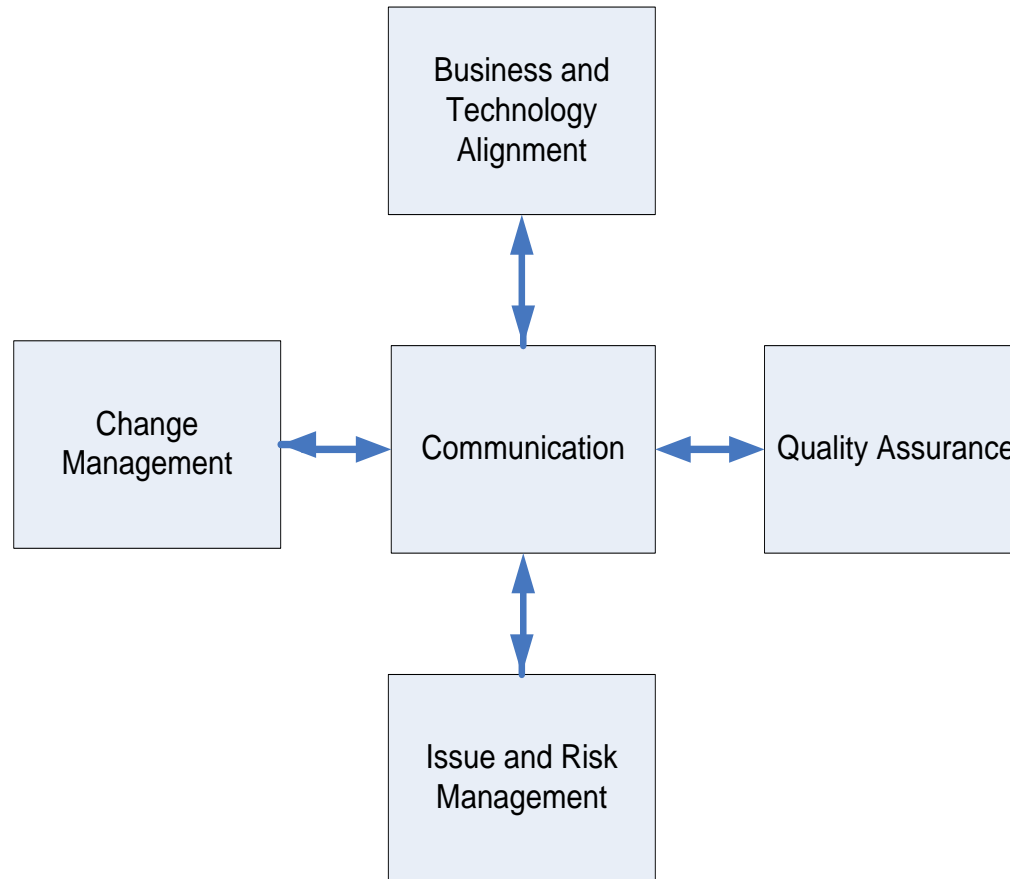
- The Project Close phase is very important from an organizational learning perspective. The Post-Mortem deliverable will provide a summary of “what went right”, and “what went wrong”. From these summaries recommendations for improvement can be made to the organization that can feed back into Lifecycle and the organization.
- It is important to also conduct mini iteration close sessions so the lessons learned from one iteration can be input into the following iterations.

## A note on Project Iterations



- **The project must produce deliverables within every iteration. Each iteration performs some level of analysis, design, construction, testing, and deployment activities. In a typical project, many of these activities will overlap in conjunction with overlapping iterations. For instance, developing user documentation is a Delivery Phase deliverable, yet can be started as soon as Component Blueprints are complete, a Solution Discovery Phase deliverable. Deliverable dependencies and delivery optimizations need to be analyzed given a project's specific situation. This same principle applies equally to all other Process Models.**
- **Although project phases will overlap, the Vision Phase and Initiation Phase should be completed before the other phases begin.**

# Four Cornerstones of Successful Projects



# Change Management Cornerstone



- **Change Management**

- Changes happen on every project and can negatively affect the project performance and client satisfaction if the Project Team does not manage them appropriately. The Project Team must build tolerance to change in the project plan and key stakeholders must agree on the process to be followed to manage (control) the changes. Uncontrolled change presents a serious risk to every project.

# Risk and Issue Management Cornerstone



- **Issue Management**

- This forms the project to-do list with sufficient information to be able to review project performance, establish trends, determine preventive/corrective actions. Through this the project manager is able to assign issues, track their progress, and measure effectiveness at dealing with issues.

- **Risk Management**

- Lifecycle provides a standard risk management plan and a risk register template with which to track and document project risks. The first formal risk analysis is normally conducted during the project kick-off meeting. Subsequently, it is the Project Team's responsibility to maintain a suitable awareness of the potential risks, and to be prepared to mitigate the most severe, as the risk picture continuously evolves.

# Risk Management Plan



- **Identify**
  - Determining which risks might affect the project and documenting their characteristics [PMI, 2000].
- **Evaluate**
  - Combination of qualitative and quantitative analysis to determine the risk's severity. This includes determining the impact, likelihood and cost.
- **Mitigate**
  - Several options are available to the Project Team to plan a mitigating response at the project planning stage.
    - *Avoidance* means to change the project to eliminate a specific risk.
    - *Transferring* the risk means to share or assign total responsibility for managing the risk to a third party (a supplier or client).
    - *Mitigation* consists of implementing actions to reduce the probability and/or impact of a given risk event below a desired threshold
- **Monitor/Adjust**

# Business and Technology Alignment Cornerstone



- **The processes and deliverables of LifeCycle are key to ensure that all the deliverables add value and are aligned with the business objectives of the project. Equally important is aligned with long term technology and architectural value. In short, LifeCycle ensure that the solution is scalable, extendable, expandable, flexible, adaptable, and high performing from both a business and technology point of view.**

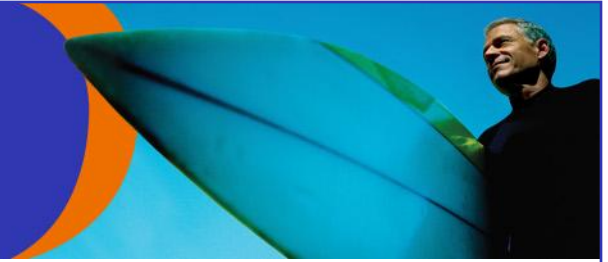
# Quality Assurance Cornerstone



- **Quality Assurance**

- Quality assurance is integral to Lifecycle. The quality assurance activities are designed to ensure/verify that the project satisfies the customers needs, and that its output are free from critical defects: this applies to any project.
- A key component of each project charter is the quality plan and acceptance criteria. This plan identifies:
  - Applicable software specification or scope documents
  - Key solution performance specifications
  - Quality targets for solution acceptance
  - Specific quality assurance roles and process expectations
  - Quality assurance reviews/milestone dates

# Quality Assurance Cornerstone



- ***Schedule Quality Assurance***

- This type of quality assurance is the responsibility of the Project Manager. This quality assurance involves ensuring that the scheduled tasks are completed on schedule. This quality assurance also involves escalating issues if the schedule quality is compromised.
- **Escalation Path:** Client Project Manager
- **Method:** Status Reports, Status Meetings

- ***Functional Quality Assurance***

- This type of quality assurance is the responsibility of the Application Architect. This quality assurance involves ensuring that the scheduled tasks are completed with the correct functionality as prescribed. This quality assurance also involves escalating issues if the functional quality is compromised.
- **Escalation Path:** Project Manager
- **Method:** Periodic Reviews of builds and deployments

- ***Technical Quality Assurance***

- This type of quality assurance is the responsibility of the Technical Architect. This quality assurance involves ensuring that the scheduled tasks are completed with the correct technical implementation as prescribed. This quality assurance also involves escalating issues if the technical implementation quality is compromised.
- **Escalation Path:** Project Manager
- **Method:** Periodic Reviews of checked in code.

# Quality Assurance Cornerstone



- ***Unit Testing Quality Assurance***
  - This type of quality assurance is the responsibility of the developers. This quality assurance involves ensuring that the components are completed with no defects that could be detected in unit testing.
  - **Escalation Path:** Project Manager
  - **Method:** TDD or Unit Testing Methodology
- ***Post Unit Testing Quality Assurance***
  - This type of quality assurance is the responsibility of the project team. This quality assurance involves ensuring that the components are completed with no defects.
  - **Escalation Path:** Project Manager
  - **Method:** LifeCycle Testing Methodology

# Quality Assurance Cornerstone



- **Key Quality Assurance Phases/Activities**

- Test Planning

- Planning for software testing elaborates the strategy for ensuring that the project's processes are capable of delivering the expected output (validation) and for verifying that the process output meets the requirements and the established quality criteria.
- Although not all the following elements will be part of every test strategy, they should all be considered when developing the project's test plan.
  - Acceptance Criteria
  - Test Methodology – how will the tests be carried out, by whom, what tools will be used, how will the tests be recorded, roles and responsibilities.
  - Test Cases – Scenarios with their objectives
  - How will code be tested? How will documentation be tested?
  - Process to deal with test results; successful test vs. unsuccessful test.
  - How to test for visible output (user-observable) and invisible output (not user-observable).

# Communication



- **High Bandwidth communication is essential to a project's success. This ensures that the proper people are informed and everyone has the necessary information to efficiently complete their tasks. Although not mandated, LifeCycle recommends strongly the following communications:**
  1. Rigorous Kick off meetings and process (internal and external)
  2. Initial and ongoing risk management meetings
  3. Daily 15 minute stand up meetings
  4. Visual Project Management as part of a SharePoint site
  5. Weekly Status Reports and Project Manager meetings. These should discuss Change Management items and Issues at a minimum)

# Break



- **Coffee and 15 minute group discussion**
  - What was the best project you were on from a solution point of view?
    - What made the solution good?
    - How could it have been even better?
    - Did the clients appreciate the architectural solution?

Questions?

