

---

# Chartering For Project Starts & Recoveries

---

Bob Galen

RGalen Consulting Group, LLC

[www.rgalen.com](http://www.rgalen.com)

[bob@rgalen.com](mailto:bob@rgalen.com)

---

# Outline

- Introduction
- Project Chartering as a *Starting* mechanism
- Alternative Examples
  - TSP – Team Software Process
  - Crystal - Exploratory 360°
- Project Chartering as a *Recovery* mechanism
- Close

---

# Introduction

- I consider establishing a Charter as the first step in any project.
- But it can also be an iterative process of Chartering
  - As discoveries are made, as problems are encountered, as scope changes

The Chartering process can be re-initiated to reestablish your goals and direction

---

# Introduction

- From a PMI perspective, this presentation extends the one time Charter exercise as part of *Initiating* to also cover iterative –
  - *Planning* – Scope, WBS, Resources
  - *Executing* – Change implementation
  - *Monitoring & Control* – Change control
  
- Key processes include:
  - Scope, Change & Risk Management

---

# Introduction

- Point is – Why can't we use the essence of establishing a Charter
  - The focus areas
  - The activities
  - The processbeyond Software Project initiation?

I think we can...

---

# Chartering

---

# Chartering - Defined

## What is it?

It's a *Process* and *Artifacts* that:

- ❑ Establishes the vision state for the project
- ❑ Defines key goals & requirements
- ❑ Captures and sets customer expectations
- ❑ Defines project participants and their roles
- ❑ Defines limits and constraints
- ❑ Establishes all resource needs and overall cost targets
- ❑ Creates a high level view to the WBS and schedule
- ❑ Initiates negotiation and tradeoffs
- ❑ Ultimately defines success

---

# Chartering – Another View

*A charter is a central document and a set of supporting documents that defines the purpose, nature and characteristics of an about to be undertaken software project.*

*It is typically constructed early in the project lifecycle, hopefully before the project is staffed and the business is pushing for a delivery date. It is usually created collaboratively as a team and shared with stakeholders upon completion.*

*It is intended to clearly set the stage for the project—aligning the team and setting goals and expectations.*

---

# Charter Components

- **Purpose**
  - Primary rational, compelling and clear, mission and vision for the effort
- **Goals**
  - Technical, business, product, and team objectives
- **Scope**
  - Customer needs, requirements, bounds
- **Organization**
  - Executive / stakeholder, project, functional organization structures
- **Resources**
  - Space, equipment, people, skill sets & capabilities, collaboration support, tools

---

# Charter Components

- **Approaches**
  - Strategy, methodologies, processes, tools, & techniques
- **Priorities**
  - Ordering, importance, trade-offs, relative to other projects
- **Assumptions & Constraints**
  - Restrictions, limits, bounds – team, process, product, & schedule
- **Risks**
  - Top n risks, known, previous history, uncertainty elaborated
- **Signoff**
  - Stakeholders

---

# Chartering Process

Beyond the documents themselves is the process for generating them. Basic steps include –

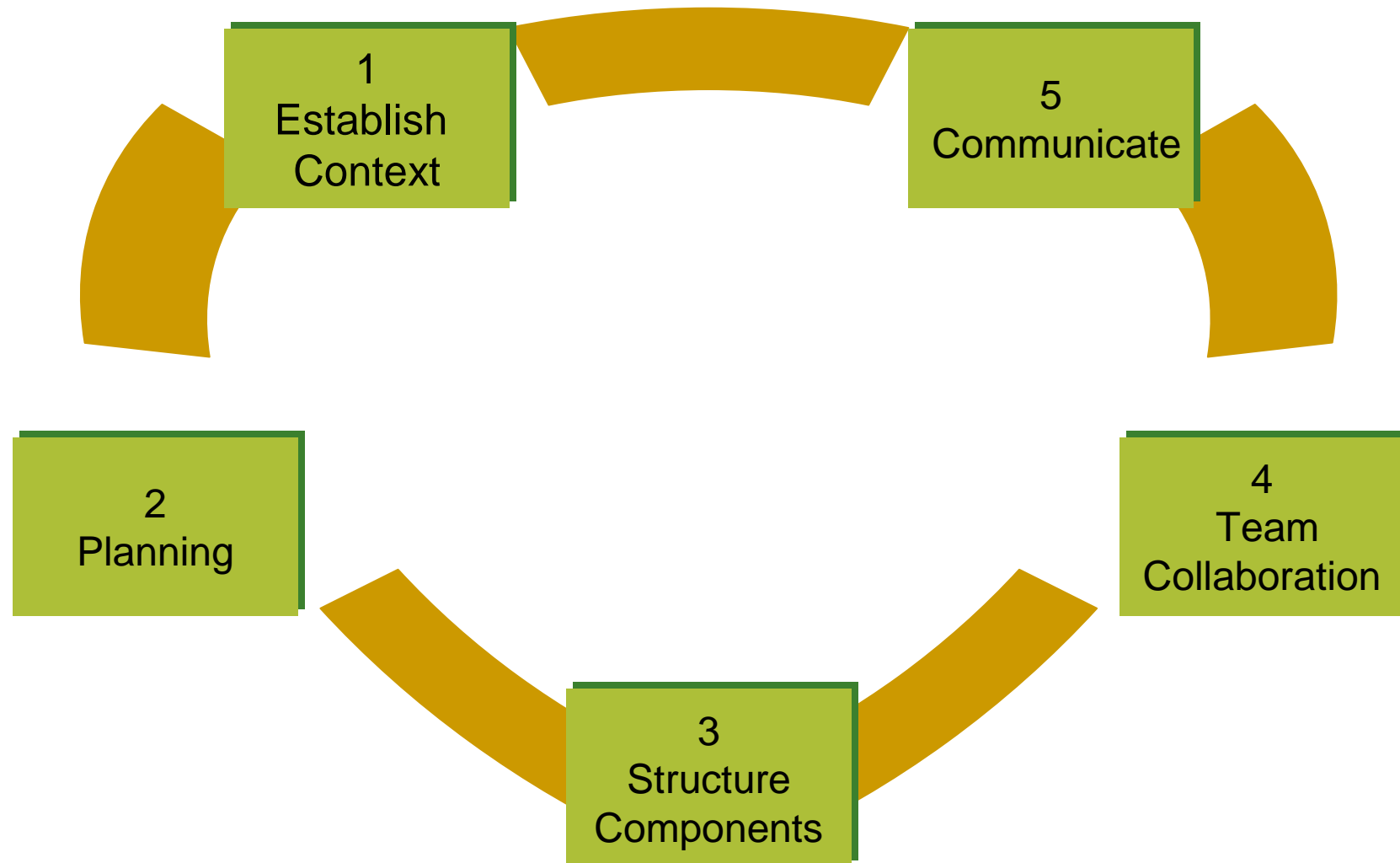
- 1. Establishing Mission, Vision, Purpose and Goals**
- 2. Team Formation, Define Roles & Responsibilities, Mapping Skills**
- 3. High-Level Planning, Risk Analysis, Create Schedule**
- 4. Stakeholder Support, Management Understanding and Team Buy-in**

# Chartering Process Flow



---

# Iterative Chartering Process



---

# Chartering – 3 Primary Contexts

- In the *Beginning*
  - To establish the initial direction, to generate momentum, to create commitment
  
- Whenever there is *Directional Confusion*
  - To re-establish the direction – goals, mission, and requirements because some part of the context has changed
  
- As a project *Recovery* mechanism
  - Based upon some sort of failure, chart the steps towards a recovery

---

# Chartering – More Subtle Outcomes

- Team + Stakeholder = alignment & shared commitment
- Shared understanding of requirements and critical success factors
- Input for rewards, recognition, and performance evaluation
- Guidance for the planning focus – level & direction
- Corrective actions and risk analysis

---

# Charter Planning

- Iterative & minimalist approach
  - Consider notions of Just Enough and Just In Time
- For each component of the Charter consider whether:
  - It's required for this context?
  - If so, do you need partial or full exploration?
  - Define exactly what is required?
  - Try to leverage previous contexts or existing information
  - Who needs to be involved?

# Charter Planning Tool

Charter Component	Project Type		
	New or Green Field	Maintenance or Recurring	Corrective Actions
<b>Purpose</b>	Always	Focused	Very focused
<b>Goals</b>	Always	Focused	Very focused
<b>Scope</b>	Always	Focused	Very focused
<b>Organization</b>	Always	As required	Usually not
<b>Resources</b>	Always	As required	Usually not
<b>Approaches</b>	Always	Always	Always
<b>Priorities</b>	Always - Broad	Focused	Very focused
<b>Assumptions &amp; Constraints</b>	Always	As required	As required
<b>Risks</b>	Always	As required	Always
<b>Sign-off</b>	Always	Always	As required

---

# Charter Planning Scenario

- Hand-out w/3 sample chartering scenarios for a sample project sequence
- We'll explore #1, next within an initiate project context
- And #3 later as part of a recovery context

---

# Chartering Scenario #1 - Context

- **Maintenance Release for Enterprise Level News and Information Portal**
  - A major release has just occurred for our Enterprise level news application which leverages (Java, EJB, and Websphere) technologies. It was quite successful, but because of schedule pressure, a few features were held in reserve for a later maintenance release that now needs to be completed. There are also a set of cosmetic “branding” changes that need to be made along with a significant number of bug fixes.
  - Of course, the business expectation of this release is to complete it in 3 weeks and to not have it interfere with preparation for the next major release. There is a single product development team, so multitasking will be occurring between the maintenance and next major release resources.

---

# Chartering Scenario #1 - Areas

## ■ Charter Planning

- No need for – Purpose and Organization
- Short focused effort for – Goals, Resources, Approaches, Priorities and Sign-off
- Full effort for – Scope, Assumption & Constraints, and Risks

---

# Chartering Scenario #1 - Focus

- In the chartering sessions, we'll focus on the following assessment & planning steps –
  - Establish the specific scope of the effort (small feature list, verify feature understanding, bug list, clarity around re-branding effort)
  - Quickly develop a plan to either support or deny the scope vs. project schedule target.
  - Identify and manage the few relevant risks for this focused effort
  - Review the results with key management and stakeholders.  
*Note: if we can't support the timing expectations, we should probably bring 2 views to the table – what percent can we deliver within the timeframe and how long to deliver 100% of the feature and defect set*

---

# Team Software Process - TSP

---

# TSP – A Chartering Example

- Team Software Process created by Watts Humphrey
  - Software Engineering Institute – SEI process initiative
  - Complements (wraps) PSP teams
- TSP defines a *launch* process that provides another framework for chartering
  - Initial launch
  - Iterate every 2-3 months, post-mortem and re-launch
  - 3-4 day exercise to define project iteration (Charter)

---

# TSP – Launch Steps

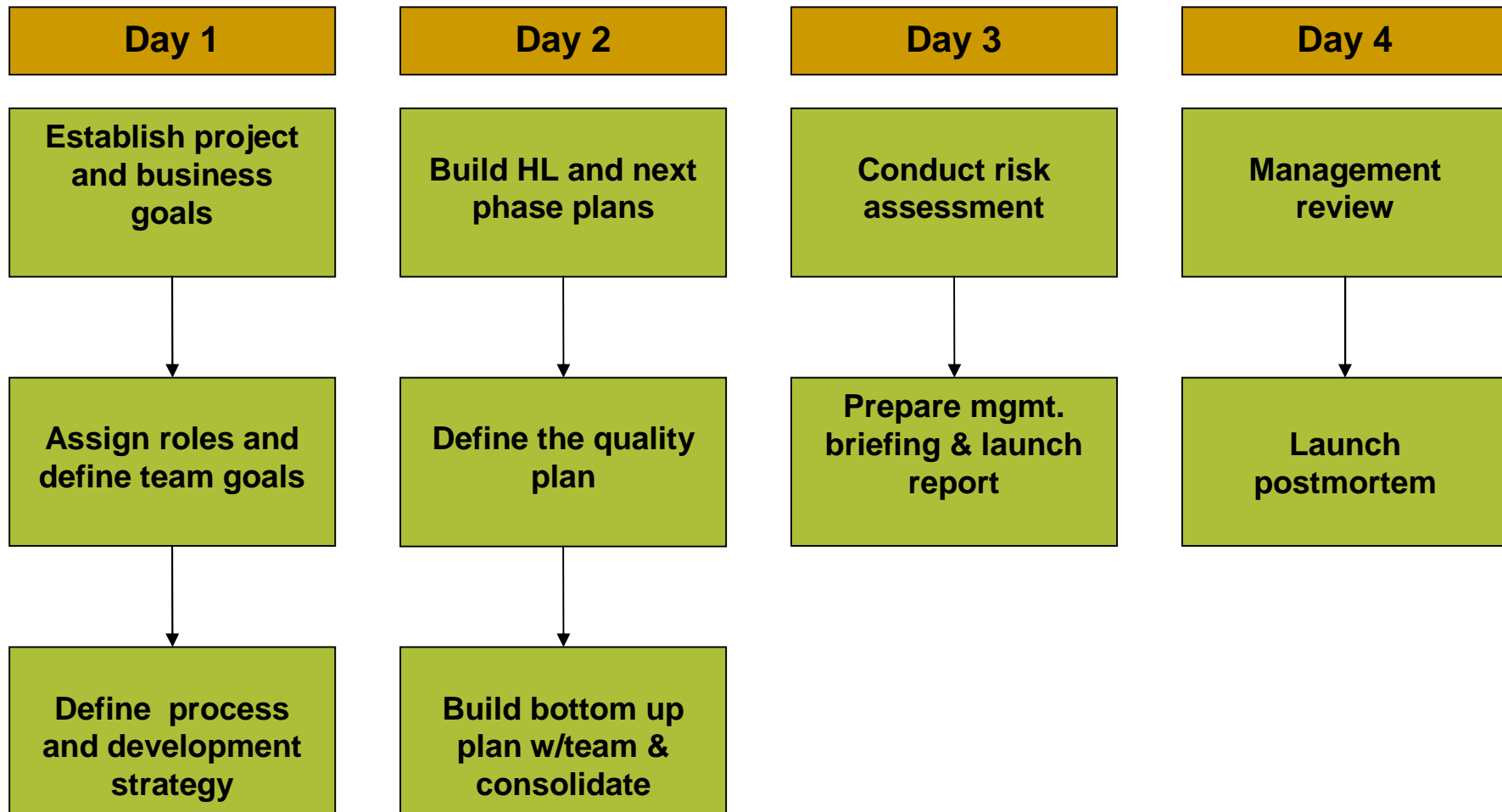
1. Review project objectives with Management, agreeing and documenting team goals
2. Establish clear team roles
3. Define the teams' development process
4. Define a quality plan and set quality targets
5. Define plans for project support facilities (equipment, labs, tools, etc.)
6. Define a high level development strategy
7. Define (working within the team) a high level, overall project plan

---

# TSP – Launch Steps

8. Defined detailed working plans for each of the project engineers (2-3 month horizon)
9. Merge the individual plans into the overall project plan
10. Rebalance the team workload to achieve a minimum overall schedule
11. Assess project risks and assign risk tracking to team members
12. Review goals, plans, schedule and risks with management
13. Start working...

# TSP – Launch Steps



---

# TSP – Core Notions

- Having an **experienced coach** lead the launches and influence team dynamics
- Using an **iterative launch process** as a means of establishing early high level plans
- Gaining management and team **buy-in before actually starting** the project
- Providing **management vision into the specifics** of what the team can actually deliver by a specific date
- Wrapping all of the above into iterations with **Postmortem and Re-launch** phases

---

# Crystal – Exploratory 360°

---

# Exploratory 360° – Attributes

The following areas are explored –

- Business Value
- Requirements
- Domain Model
- Technology Plans
- Project Plan
- Team Makeup
- Methodology or Working Conventions

---

# Exploratory 360° – Dynamics

There is an aspect of *sampling* that occurs in the exploration to –

- ❑ **Sample requirements** – course level use cases developed, can we define the system?
- ❑ **Sample models** – can we define the system?
- ❑ **Sample business value** – to determine raw scope, schedule and value feasibility. Does it makes sense?
- ❑ **Sample technology** – can we connect things? Do we understand the technology?
- ❑ **Sample testing** – can we sufficiently test it? Do we have the tools for the technologies? The skill level?

---

# Exploratory 360° – Core Notions

- Explore over a series of days or 1-2 weeks
  - Depending on the challenges and extent of the project / system
- Very much a technology-driven methodology with emphasis on –
  - Modeling & Design
  - Use case development
  - Iterative, skeletal construction
- It's very focused on **technical feasibility**, understanding, implementing and **confirming business value**

---

# Chartering as a Recovery Tool

---

# Chartering as a Recovery Tool

- Course grained to fine grained
- Very focused
- Typically look towards
  - Scope & Resource implications
  - Risks – very targeted towards the problem at hand (looking for root cause)
  - Sign-off for stakeholder notice and tradeoff negotiation
- Final step is re-establishing the Charter with the team and stakeholders

---

# Chartering Scenario #3 - Context

- Major Release for Enterprise Level News and Information Portal, Problems!
  - Fast forward again from the above scenario... The team has already completed the chartering efforts for the entire project. We all felt that there was too much content, so we committed to completing 80% of the features and still holding to the 4 month, 4 iteration schedule. Our business stakeholders felt that this level of delivery would be sufficient to attack the market opportunity and that a follow-up release could complete the effort.
  - We're now just approaching the end of the second iteration.
  - But problems are cropping in. The first iteration acceptance tests were only 50% completed. Root cause here was primarily business resource conflict, but there were insufficient testing resources applied to the tasks as well. At the end of the second iteration, it appears that there is no preparation or resource availability for newly introduced feature testing.

---

# Chartering Scenario #3 - Context

- ❑ In addition, preliminary backend architectural development work has uncovered some performance problems when interfacing to the database. It's unclear where these problems specifically lie, but overall system performance is only a fraction of where it needs to be for the new content.

## ■ Charter Planning

- ❑ No need for – Purpose, Goals, Organization, Approaches, Priorities, Assumptions & Constraints
- ❑ Short focused effort for – Risks and Sign-off
- ❑ Full effort for – Scope and Resources

---

# Chartering Scenario #3 – Focus

- In the chartering sessions, we'll focus on the following assessment & planning steps –
  - Including key architects, senior developers and backend database experts in a problem analysis and resolution planning session. We need to sort out root cause of the problem and its relationship to architectural evolution / effort.
  - Contact the business team to understand the impediments to creating solid feature acceptance tests – on time. This needs to be improved and fed into the testing team for execution. This is a critical, progress blocking issue.
  - Contact the QA manager to understand resource constraints and other factors to their being behind schedule. Get to root causes, future risks and mitigation possibilities. Take this into the re-planning part of the chartering.

---

# Chartering Scenario #3 – Focus

- Including the entire team, re-plan the final 2 iterations. Since scope is well understood, a good position indeed, we only need to re-map work across the available time, including test time reassessments.
- Review the re-plan results with key management and stakeholders. *Note: we should probably bring 3 views to the table (delivering 100% of the features, delivering the previously negotiated 80%, and what percentage can be delivered in the final 2 iterations supporting the original 4 month target)*

---

# Chartering Wrap-up

- Define a Charter template for your organization
- Wrap it with an iterative model
- Adapt chartering activity to the situation
  - Just in time and just enough
  - Specific context
- Recovery is about
  - Understanding where things stand and constructing a way forward
  - Adjusting existing plans based on discovery & re-planning

---

# Thank you for taking the time!

## Questions?

References: [www.rgalen.com/estimating.html](http://www.rgalen.com/estimating.html)

### Contact Info:

RGalen Consulting Group, L.L.C.

PO Box 865, Cary, NC 27512

919-272-0719

[www.rgalen.com](http://www.rgalen.com)

[bob@rgalen.com](mailto:bob@rgalen.com)