

# Project Manager's Roadmap

**"We're all smarter together"**

## Think Top Down!

Methodology	<b>Define</b> <b>Plan</b> <b>Execute</b> <b>Close</b>
Checklists	<b>Conflict Resolution Modes</b> <b>Contract Outsource Management</b> <b>Mentoring Basics</b> <b>Supplementary Resources</b>

**Success Equation: Scope divided by  
Resources, Time, Technology, and Processes**

Proactive Project Manager integrates business need, user requirements, technology, budget, collaborative team/contributor efforts, and processes into an effective solution

### **DEFINE**

#### **Initiate Project Activities**

##### *Build business case:*

- Need or opportunity
- Mission and purpose
- Scope and deliverables
- Objectives, assumptions, and constraints
- Timelines/major milestones
- Budget, costs, benefits, and payback (return on investment (ROI))
- Specifications/quality
- Major risks
- Project team (organizational structure and skill sets)

##### *Define:*

- Customer (external or internal)
- Sponsor, stakeholder(s), and contributors
- Objectives (clear and realistic)
- Strategy and tactics
- Scope and methodology
- Budget
- Organization (projectized, matrix, or functional)
- Core team members (select, evaluate, and monitor)

# MoneyWords.Com Checklist

## PLAN

### Prepare for Success

#### *Estimate:*

- Assess planning factors
- Evaluate personal/team experience
- Review and select estimating methods
- Investigate historical data and methodologies
- Size (lines of code, function point, or feature set)
- Estimate effort (man-months or FTEs)
- Document assumptions and estimates
- Schedule (calendar months)
- Estimates review and sign off

Tip: Recalibrate after first phase (or when preliminary analysis is complete)

#### *Conduct initial risk assessment:*

- Project team (technology and methodology skill level, business knowledge, team composition, location, and environment)
- User involvement and knowledge
- Organizational impact and priority
- Senior management commitment. READ "motivation and intent"
- Project attributes (development technology and methodology, schedule constraints, project priority and novelty, and project development cycle and implementation)
- Requirements stability
- Stakeholders
- Criticality, size, and complexity

#### *Prepare:*

- Project charter
- Roles and responsibilities matrix

Tip: Consider project size and complexity, product availability and need, and level (quality) of customer involvement

#### *Kick-off:*

- Assign team members
- Hold kick-off meeting (agenda, location, props, and minutes)

#### *Buy-in:*

- Brief project team and contributors
- Obtain commitments from management, team, and contributors

#### *Tools:*

- Identify
- Document and install
- Train team members

#### *Metrics protocol:*

- Review project objectives and scope
- Identify critical success factors (CSF)
- Identify metrics
- Balance and prioritize metrics
- Set targets (knowledge and achievement goals)

#### *Metrics (starting point):*

- Cost and resource data
- Number of changes and defects
- Process conformance
- Productivity
- Code changes and growth
- Requirements changes

## MoneyWords.Com Checklist

- Use cases (completed, projected, integrated, and tested)
- Schedule (development phases, milestones, activities, total effort, estimate time to complete, establish baseline, variances, and deliverables)
- Issues
- Code builds
- Risks
- Quality metrics
- Lines of code
- Functioned points

Tip: Apply metrics to control, evaluate, and improve performance

### Document Activities

*Generate:*

- Work Breakdown Structure (WBS) (work units)
- Define WBS activity levels (phases, subphases, and tasks)
- Review WBS with project team
- Finalize WBS elements

*Organize:*

- Project notebook
- Server space (activate backup procedures)
- Web space (activate backup procedures)

*Clarify:*

- Project objectives after preliminary requirements document is available
- Assumptions and dependencies
- Requirements (scrub)
- Development activities are traceable to requirements
- Project specifications (generate)

*Describe:*

- Entry and exit conditions
- Review and approval criteria
- External project dependencies

Tip: Tailor all activities to project, document tailoring rules, and seek appropriate approvals

### Generate Software Development Plan (SDP)

- Refer to **Software Development Plan** for narrative

Notes:

1. Recognize planning for what it is...A RESULTS-ORIENTED PROCESS
2. Go for the jugular! READ "Ship Date!"
3. Knock it apart, then put it back together...ANATOMIZE, THEN SYNTHESIZE
4. ALL activities should support methodology and GOAL
5. RELEVANT approximations are better than inconsequential minute estimates
6. Prepare ONLY plans that can be MONITORED
7. Plans identify activities, but PEOPLE MAKE THEM HAPPEN

### Prepare Schedule(s)

*Top-down and bottom-up process:*

- Identify scheduling tool
- Identify deliverables and milestones
- Conduct planning meetings
- Get project team inputs
- Scrub data inputs
- Generate schedule(s)

## MoneyWords.Com Checklist

- Review schedule with team
- Run schedule and update
- Obtain schedule approval
- Distribute approved schedule(s)

Tip: Prepare summary schedule for management reviews and micro schedules for day-to-day activities by phase. Schedule complex components first

### Establish Key Processes (Best Practices)

*Activate:*

- Requirements and issue management
- Project planning
- Tracking and oversight
- Configuration/change management
- Quality assurance
- Outsource management

Tip: Enforce coding standard and software integration procedure. Key results areas are customer, business unit, department, and project team members

## EXECUTE

### Drive to Completion

*Power sources:*

- Formal
- Reward
- Penalty
- Expert
- Referent

*Prioritize project manageables:*

- Time
- Cost
- Quality

*Control scope:*

- Gain product understanding and customer agreement
- Baseline customer's requirements
- External changes (marketing, user driven, or environment)
- Internal changes (no gold-plating - meet minimum requirements)
- Impact analysis (budget, schedule, and risk)
- "Fix it in the next release" (establish sustaining engineering bin)
- Negotiate changes
- Clear go/no agreement

*Manage risks:*

- Identify
- Document
- Evaluate
- Prioritize
- Mitigate

*Motivate team:*

- Vision and identity
- Inspire, ENGAGE, and challenge
- Manage team as a team
- Communicate, get feedback, and follow-up
- Build mutual trust (honesty, openness, fairness, and RESPECT)
- Delegate tasks to team (EMPOWER)
- Make team responsible for actions, not individuals
- Maintain productive environment

# MoneyWords.Com Checklist

## Daily:

- Focus on the GOAL!
- STOP to evaluate progress
- Avoid scope CREEP (and making major mistakes)
- Lead and solicit FEEDBACK (emphasize collaborative effort)
- THINK about relevant issues (and concerns)
- Log and ACT on issues!!!
- Troubleshoot (problem solve), and/or escalate, but resolve quickly
- MONITOR action(s) so that you get the desired results
- Make everyone a winner

Tip: The name of the game is responsive issue resolution. Actively keep sponsor, stakeholders, and contributors in the loop. After the construction phase, code build daily (heartbeat of project)

## Weekly:

- Conduct project review (focus on measures of merit)
- Track against the plan (actual progress/work outstanding/gap analysis)
- Take corrective action(s) (identify issues, recalibrate project, and document)
- Communicate upcoming events (lookahead schedule with milestones/tasks)
- Share knowledge, skills, and lessons learned
- Review metrics, productivity, quality, schedule, and cost
- Ensure key processes and practices are followed
- Monitor vendors and subcontractors
- Generate status report (performance reviews, key issues, top 10 risks, jeopardy process (OK, Alert, and Jeopardy), variance and trend analysis, earned value analysis, and communicate state of project)
- Maintain "user" involvement (critical)
- Complete deployment (and train users)
- Say "Thank you"

Tip: Conduct systematic technical reviews after each phase

## System Test

### Levels:

- Unit
- Integration
- System**
- Acceptance
- Site (large scale projects)

### Metrics:

- Staff hours burned
- Total computer hours used to date
- Source code growth
- Defects and changes (categorize by severity)
- Requirements questions, answers, changes, and TBDs
- System size, effort, schedule, and reuse
- Tests executed
- Discrepancies reported and fixed

### Tools:

- Test report checklists and forms
- Configuration management
- Software analysis tools
- Regression testing

## MoneyWords.Com Checklist

### *Entry criteria:*

- Code unit tested and integrated
- Build test plans executed
- System test plan completed
- User's guide drafted

### *Tasks:*

- Plan test
- Identify test requirements
- Assess risk
- Develop test strategy
- Identify test resources
- Create schedule
- Generate test plan
- Design test
- Conduct workload analysis
- Identify and describe test cases
- Identify and structure test procedures
- Review and access test coverage
- Build test lab (simulate actual working environment)
- Implement test
- Record or program scripts
- Identify test-specific functionality in design and model
- Establish external data sets
- Execute tests
- Execute test procedures
- Execute regression tests
- Evaluate test execution
- Verify results
- Investigate unexpected results
- Log defects
- Evaluate tests
- Evaluate test-case coverage
- Evaluate code coverage
- Analyze defects
- Determine if test completion criteria has been achieved
- Prepare for acceptance testing

### *Exit criteria:*

- System test plans successfully executed
- Acceptance test plan finalized
- User's guide and system description completed
- Configuration audits are done

## **CLOSE**

### **Mission Accomplished**

- Customer accepts build
- Hold postmortem
- Focused on lessons learned and review scope, objectives, and deliverables from business and technical perspectives
- Outline tangible and intangible project benefits
- Evaluate costs (planned/actual)
- Review all information relevant to future development
- Make recommendations
- Archive project documentation and source code

## MoneyWords.Com Checklist

Tip: Project closure activities are about learning, future development efforts, improvement of processes, and key result areas. No finger pointing! Finally, act on recommendations

### Conflict Resolution Modes

- Collaborate (problem solve)
- Compromise (sharing)
- Accommodate (smoothing)
- Withdrawal (avoiding)
- Competing (forcing)

### Contract Outsource Management

- Outsource planning (procurement resources and plan, market conditions, constraints, assumptions, SOWs, expert judgement, evaluation criteria, and documents)
- Contract solicitation (qualified vendor list, bidders conference, and Request for Proposal (RFP))
- Source selection (procurement policies, proposal screening, estimates, evaluation criteria, and contract negotiation)
- Contract administration (contract, schedule, performance reporting, reviews, results, change requests, invoices, payments, correspondence, and contract changes)
- Closure (documentation, audits, acceptance and closure)

Tip: Carefully select vendor(s) and hold to the same standard as project team members and contributors

### Mentoring Basics

*Develop potential:*

- Focus on performance, achieving results, and professional growth
- Provide penalty-free relationship
- Be a positive resource and candid
- Encourage questions, critical thinking, and self education
- Develop strengths, knowledge, and skills
- Assist in creating an individualized improvement plan

Tip: Meet with mentoree on a regular basis

### Supplementary Resources

Checkout **Best of Breed Books** at [WWW.MoneyWords.Com](http://WWW.MoneyWords.Com). Also, click **BookMarks**, then IT Project Management, Science and Technology, and Internet for more resources. Links are updated weekly

**Note:** One shoe does not fit all, a company's culture, values, project management maturity and understanding, budget, goals, and project size/complexity drive the quality and the level of its project management effort

### ABOUT TOM WELCH

To learn more about Tom Welch, click **Project Manager** and **Web Content**

### ACKNOWLEDGEMENTS

Tom Welch gratefully acknowledges the feedback of Dave W. Farthing, School of Computing, University of Glamorgan, UK