2012

Basics of Realistic Scheduling

12 Tips to Realistic Scheduling

Presentation slides and speaker notes for Basics of Realistic Scheduling course

Laura Lee Rose
www.LauraLeeRose.com
3/21/2012

A *GOTO Guy/Gal* Workbook series

Corporate Exit Strategies for the Blooming Entrepreneur

Tools for the *GOTO* Person

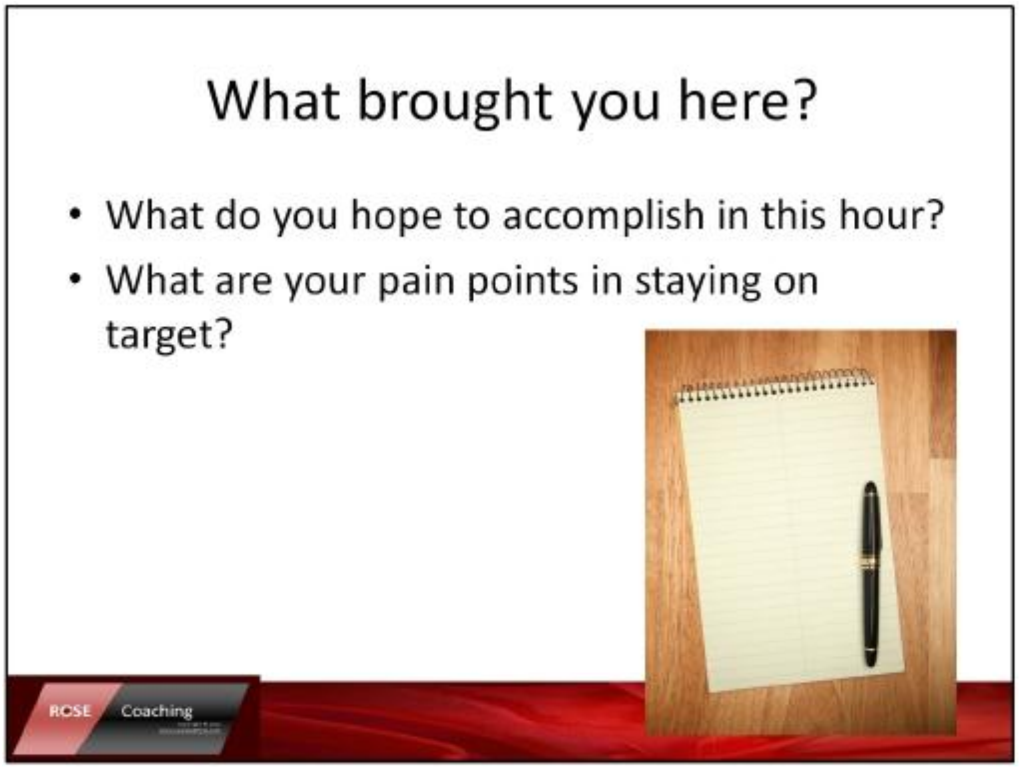
Professional Development Series

-

Speaker Notes

Welcome to today' session the basics of realistic scheduling. We cover 12 easy tips to keep us on target:

1) Don't allow "being busy" derail your commitments
2) Document detail task list
3) Identify critical paths and bottlenecks early
4) Work only on what will be of value to the customer
5) Institute customer-sponsored releases
6) Master effective and rapid decision making
7) Rigorously institute reasonable forcing functions
8) Execute effective meeting management
9) Accept progressive refinement
10) Reduce inventory where ever possible
11) Beware of heroics culture
12) Don't be the source of the chaos



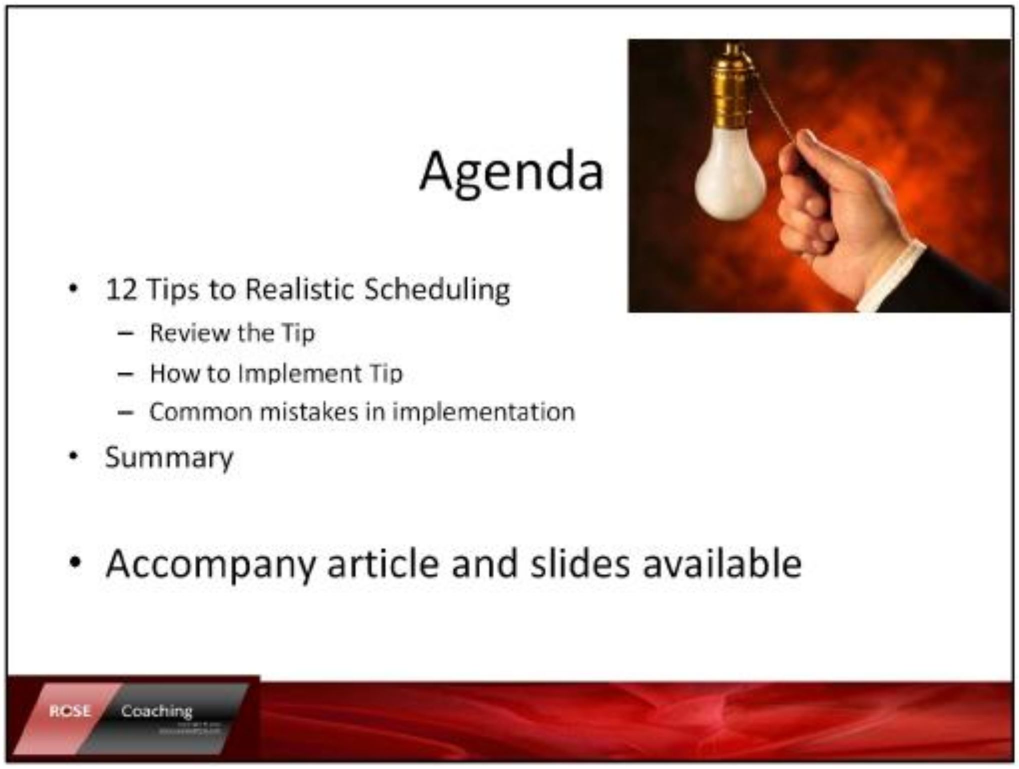
What brought you here?

- What do you hope to accomplish in this hour?
- What are your pain points in staying on target?

ROSE Coaching

In the webinar live version, there is normally interaction and Q&A.

If you have a question during your self-study, feel free to become my facebook friend at http://www.facebook.com/laura.l.rose and send me a chat question any time we're both on the computer. OR email me at LauraRose@RoseCoaching.info



Agenda

- 12 Tips to Realistic Scheduling
  - Review the Tip
  - How to Implement Tip
  - Common mistakes in implementation
- Summary
- Accompany article and slides available

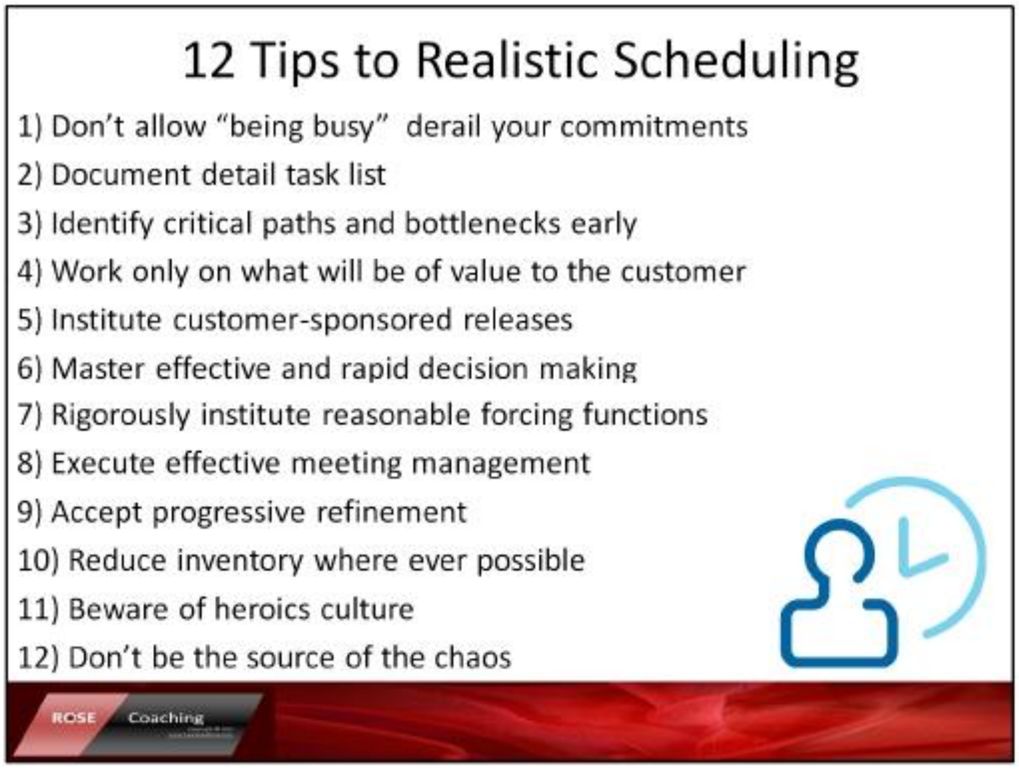
ROSE Coaching

Accompany articles can be found at www.LauraLeeRose.com

Twelve tips

Twelve tips for realistic scheduling in a software development project

Additional exercises included in these presentation notes as well as additional resources for further education.



12 Tips to Realistic Scheduling

1) Don't allow "being busy" derail your commitments
2) Document detail task list
3) Identify critical paths and bottlenecks early
4) Work only on what will be of value to the customer
5) Institute customer-sponsored releases
6) Master effective and rapid decision making
7) Rigorously institute reasonable forcing functions
8) Execute effective meeting management
9) Accept progressive refinement
10) Reduce inventory where ever possible
11) Beware of heroics culture
12) Don't be the source of the chaos

ROSE Coaching

Most people have created schedules, whether it's a timeline to drop various children off at various soccer and dance practices, or to manage large complicated software projects. Most project schedules start off with a start and end-date. They include tasks, their durations and dependencies between the tasks. But in real-life, the unexpected happen; and they involve real people who can influence, control and even complicate things. When we do not handle the unexpected and the interpersonal elements well, our schedules fall apart.

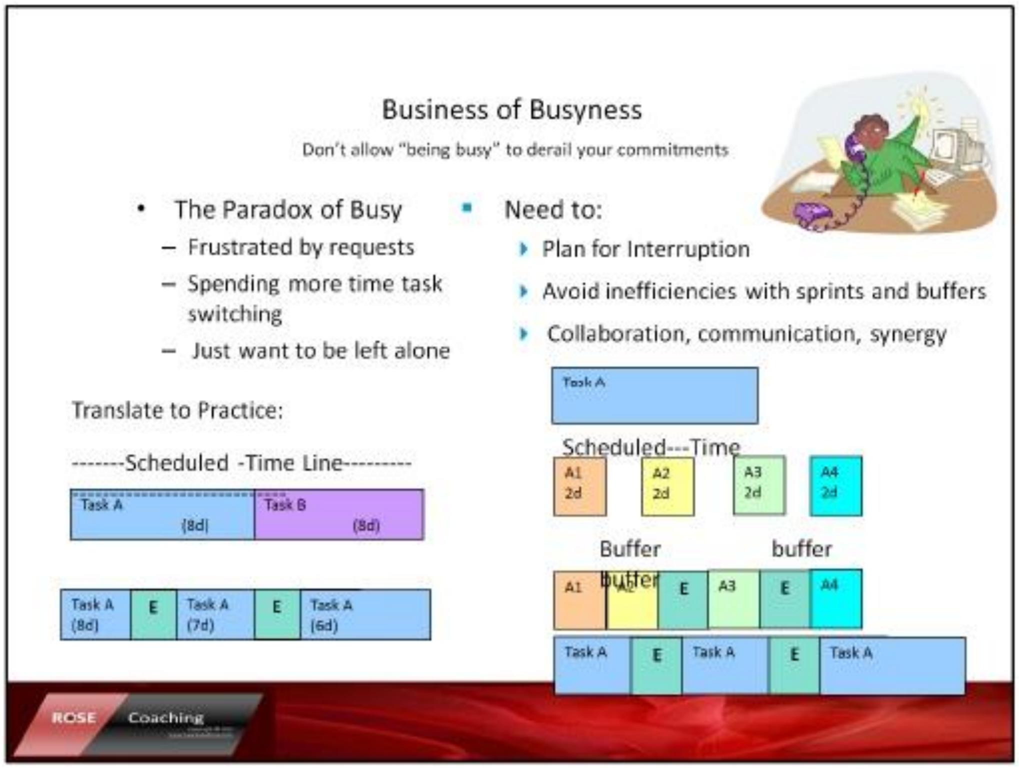
Realistic scheduling need to cover that above, and as such, is a combination of art and science. Realistic scheduling is very difficult.

Even acknowledging all of the above, there are some techniques to keep one's sanity that go beyond the notes, checklists, calendars and appointment books. These tips center on prioritization, clarifying values, and comparing the relative worth of each activity.

They combine the conventional checklists with preserving and enhancing relationships to accomplish the desired results.

- Exercise: List the various things that repeatedly take your time. As you go through this work, continually review those items in respect to the concepts that are being shared.

Speaker Notes



Business of Busyness

Don't allow "being busy" to derail your commitments

- The Paradox of Busy
  - Frustrated by requests
  - Spending more time task switching
  - Just want to be left alone
- Need to:
  - Plan for Interruption
  - Avoid inefficiencies with sprints and buffers
  - Collaboration, communication, synergy

Translate to Practice:

-------Scheduled -Time Line---------
Task A (8d)
Task B (8d)
Task A (8d)
E
Task A (7d)
E
Task A (6d)
Task A
Scheduled---Time
A1 2d
A2 2d
A3 2d
A4 2d
Buffer
buffer
A1
A2
buffer
E
A3
E
A4
Task A
E
Task A
E
Task A
ROSE Coaching

Do you find that the busier you are the more interruptions and requests you get? Do you spend more time switching from task to task than actually accomplishing something? Have you lost your patience and can't find time to talk to your team mates? Do you just want be left alone?
I call this the paradox of "busy", because, what you "want to do" and what you "need to do" are direct opposites.
In truth, the busier we are, the more patience we need to command[1].   The more in-demand we are, the more important it is to talk to people. The larger the load, the more we need to collaborate, delegate and work in teams

[1] Idle hands can easily drop everything to accommodate. Ironically we don't want idle hands. We're attracted to lean organizations (doing more with less). Therefore, we need to create an empathic, supportive business to support "busyness".

More info and tools

- Detail Learning Sessions
  - Weekly modules
- eBook
  - Slides and speaker notes

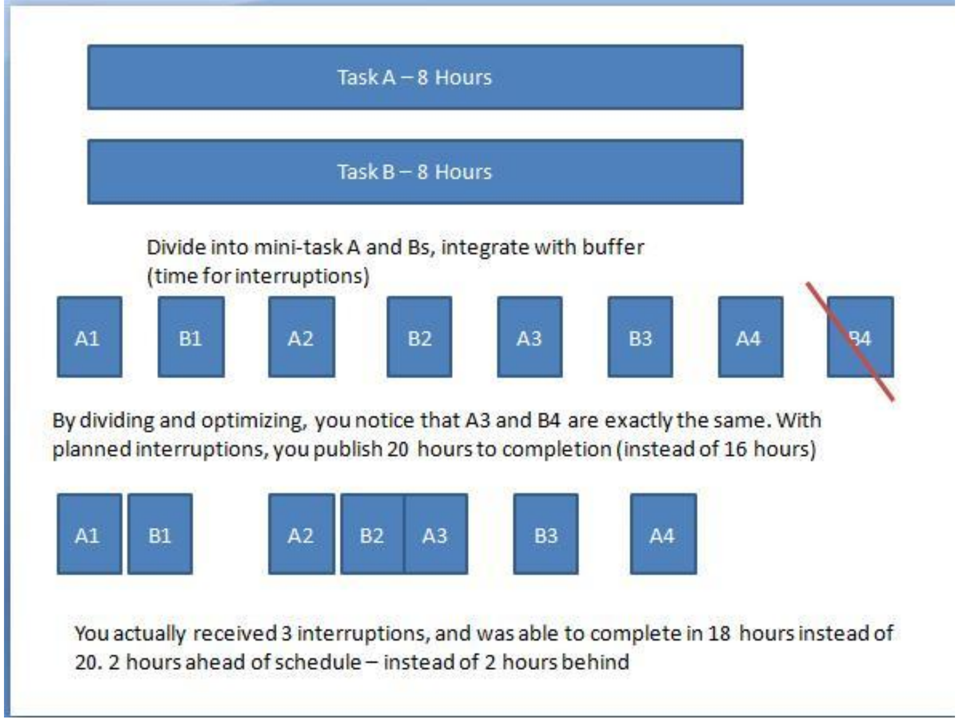
Multitasking or ‘allowing interruptions’

To me, multitasking is merely a euphemism to “allowing interruptions”.

One technique I use is the Sprints and Buffer technique.

1. Divide your tasks into smaller mini-tasks with scheduled buffer time between the tasks. This way you can make forward progress on a multiple of things — without churning from task to task.
2. When an interruption enters the queue, simply schedule that interruption at the next available “buffer time”. This allows you to complete several different categories of tasks AND handle the unexpected interruption.
3. If you are a valuable component of your company or business, then you will be interrupted…you will be in demand. That’s the definition of an MVP.

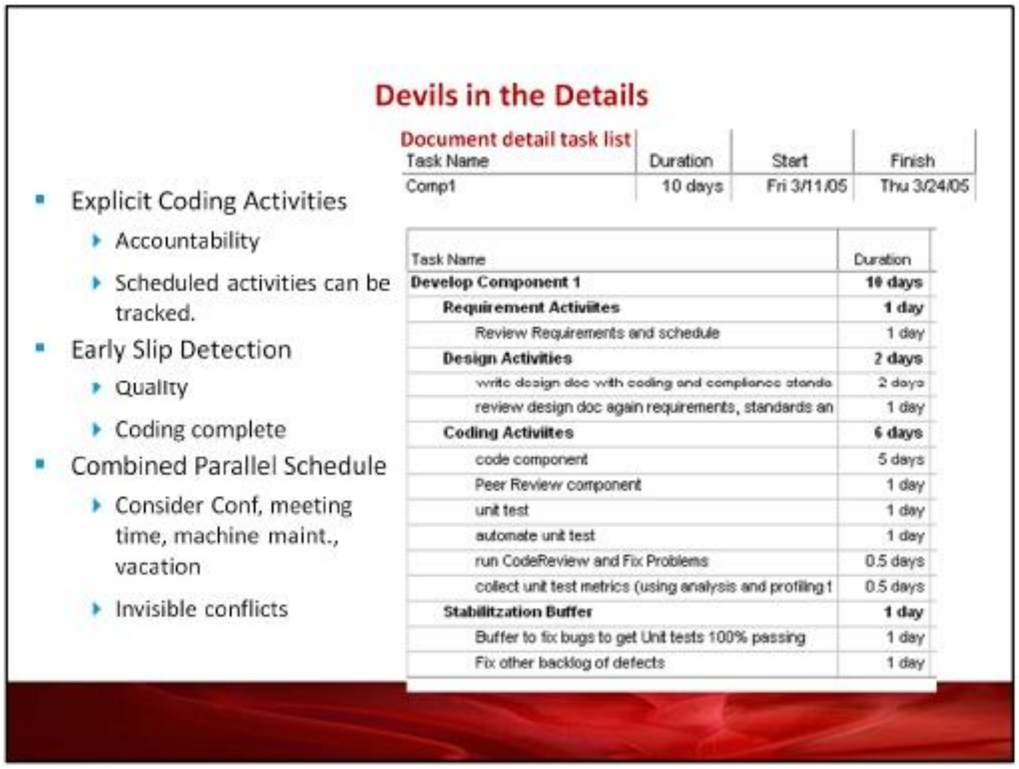
The trick isn’t to turn-off that flow of influence and appreciation; the skill is to merely manage your time better. Planning for the unexpected is the key. We don’t know what will show up — but we do know something will show up. Therefore, it makes sense to actually schedule for those inevitable interruptions.



Divide into mini-task A and Bs, integrate with buffer (time for interruptions)

By dividing and optimizing, you notice that A3 and B4 are exactly the same. With planned interruptions, you publish 20 hours to completion (instead of 16 hours)

You actually received 3 interruptions, and was able to complete in 18 hours instead of 20. 2 hours ahead of schedule – instead of 2 hours behind



Devils in the Details

- Explicit Coding Activities
  - Accountability
  - Scheduled activities can be tracked.
- Early Slip Detection
  - Quality
  - Coding complete
- Combined Parallel Schedule
  - Consider Conf, meeting time, machine maint., vacation
  - Invisible conflicts

Document detail task list

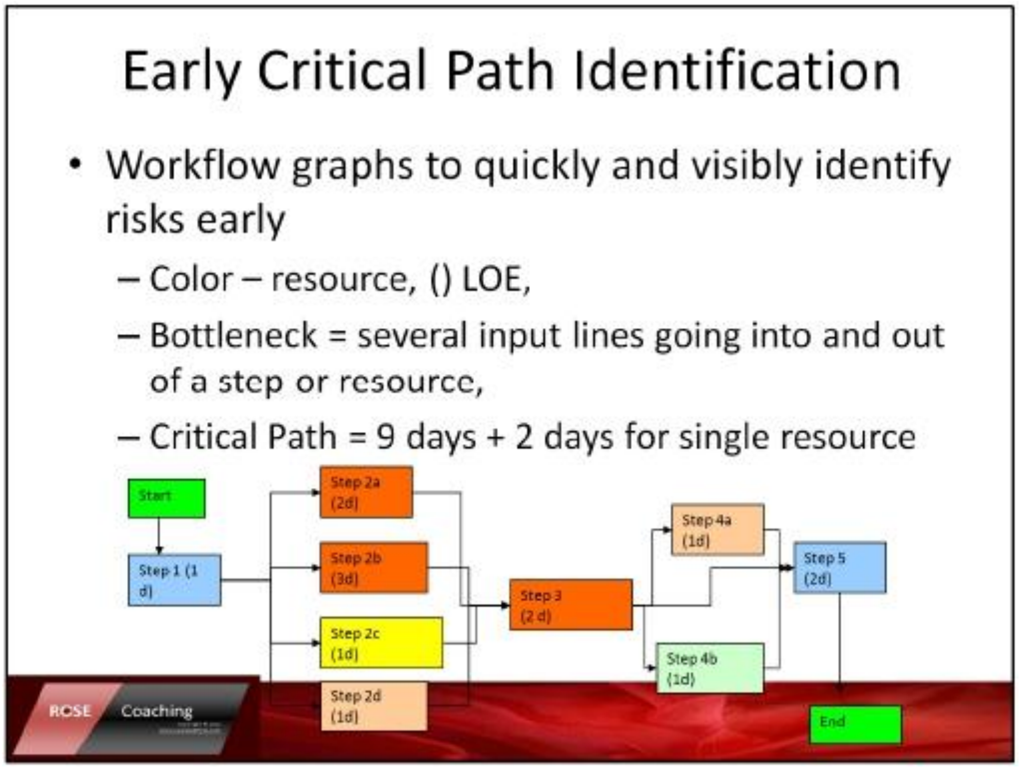
Task Name	Duration	Start	Finish
Comp1	10 days	Fri 3/11/05	Thu 3/24/05

Task Name	Duration
Develop Component 1	**10 days**
Requirement Activiites	**1 day**
Review Requirements and schedule	1 day
Design Activities	**2 days**
write design doc with coding and compliance standa	2 days
review design doc again requirements, standards an	1 day
Coding Activiites	**6 days**
code component	5 days
Peer Review component	1 day
unit test	1 day
automate unit test	1 day
run CodeReview and Fix Problems	0.5 days
collect unit test metrics (using analysis and profiling t	0.5 days
Stabilitzation Buffer	**1 day**
Buffer to fix bugs to get Unit tests 100% passing	1 day
Fix other backlog of defects	1 day

Speaker Notes

It's important to explicitly state all your coding activities, from Design Reviews, Code Reviews, Unit Test Planning/Reviews, Execution and Re-execution of unit tests, fixing defects, coding, automation….. in your schedules. If you don't explicitly make them visible, it's easier to ignore or skip them when we're running late. For instance, if we initially planned 10 days for all our coding activities, and we just illustrate "coding – 10 days"…. There is no forcing function to limit our actual coding time to allow us to do the other items on our list.
The more explicit and externally published our schedules are, the easier it is to track our progress, delays and quality initiatives.

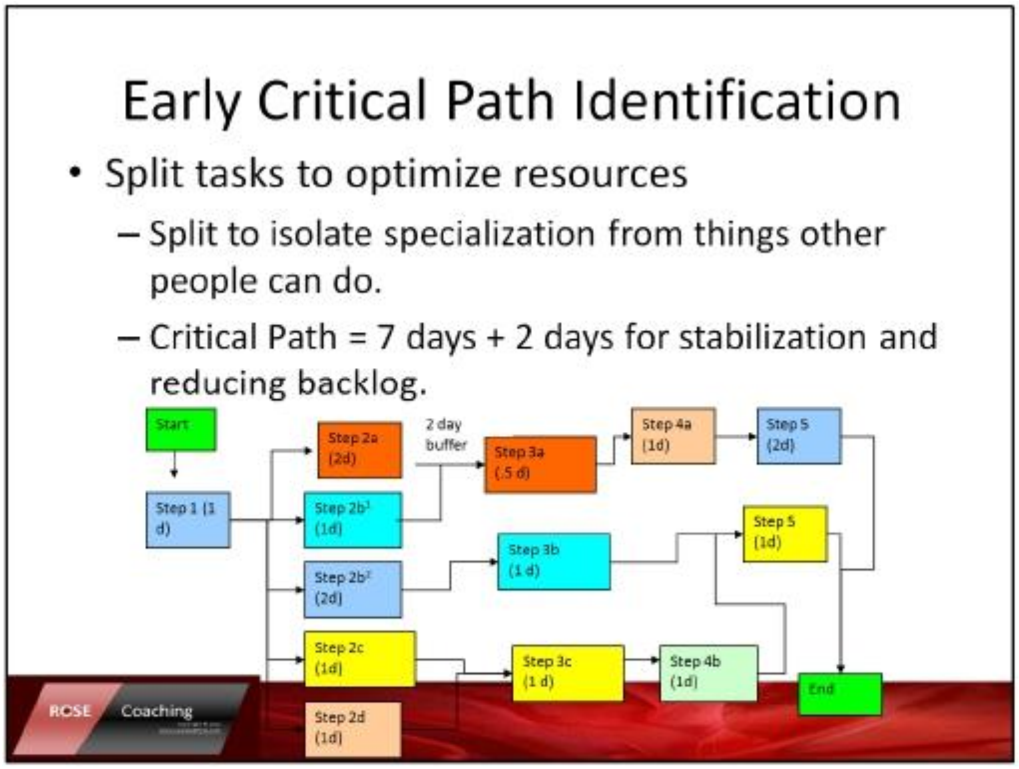
- Exercise: Review the terms that you use daily. Are you sure that your team share the exact definition?
  - What does "done", "Success", "Acceptable Quality" mean to you and your team?
  - Are you sure you are on the same page?



Early Critical Path Identification

- Workflow graphs to quickly and visibly identify risks early
  - Color – resource, () LOE,
  - Bottleneck = several input lines going into and out of a step or resource,
  - Critical Path = 9 days + 2 days for single resource

"Risk Management" has always been highly publicized as an important project management tool. Yet, we don't really take the time to model or study our workflow to identity the risks, critical paths or bottlenecks early. Like defining our level of effort, we often rely on a quick and adhoc approach; depending upon our mental review of past experiences. Very rarely does the Risk Management exercise even involve a peer review of all the tasks and workflows of the project. If we don't understand all the tasks and timing, we don't realize the majority of the risks. If you don't realize the risks, you can't manage them.
A very effective and easy way to quickly and visibly identify risks in a project is to outline the process workflow in a flowchart. The workflow method can be used to analyze anything. For instance, workflow analysis is effective on component dependencies, process step dependencies and even resource conflicts. Take the below diagram of a process. The below outlines the different components and the colors are used to illustrate the different resources doing the activity, while the estimated duration is in ().
Without resource considerations, the length of time through the critical path is 9 days (longest time through the various sequential steps). But because we've single sourced Step2a, Step2b and Step3, we need to add an additional 2 days because even though the steps are not depending upon each other, the resource on those steps are. We are now up to 11 days.
Continuing with our analysis, if there are several input lines going into and out of a step or resource, you have visually identified an architectural or structural bottleneck. In this simple example there are multiple items dependent on that Step3; therefore, we have a real bottleneck not only in the resource but in the architecture. Unless Step2a, 2b,2c,2d all completed at the right time, Step 3 can't be done. If the resource on Step 3 is stuck on Step2b, we are completely blocked. No other steps can be started. This makes the orange resource in the critical path. If we wait until the teams have started coding, and we actually hit the bottleneck, there's little we can do about it. The orange resource is already in deep. He is the only one that knows the code in Step2a, and Step 2B, and he's probably coded additional assumptions into all three steps because he's the single owner of that code. He complicated the interdependencies to make it faster to complete (once again - because he is the single owner). He is pretty much entangled, such that we can't efficiently add a resource to help him.



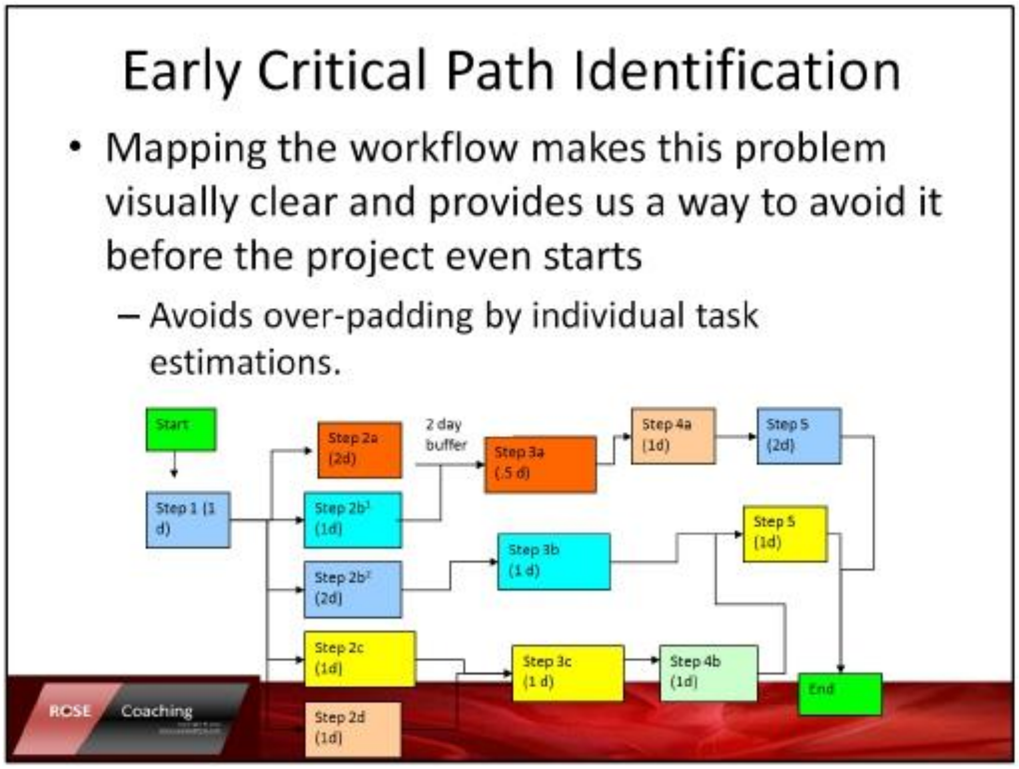
Early Critical Path Identification

- Split tasks to optimize resources
  - Split to isolate specialization from things other people can do.
  - Critical Path = 7 days + 2 days for stabilization and reducing backlog.

Once you've diagramed your initial flow, you optimize to correct the risks and bottlenecks around the dependent components and resources. In this example, although I've split my tasks into additional steps, my critical path is just 7 days (shorter than my original scheme). I still feel that resource Orange or Step 3a is a potential bottleneck, so I schedule a 2 day buffer before the potential bottleneck. This allows all the sub-steps (Step2a, 2b,2c,2d) to accumulate in a slight holding pattern. This stabilization period is a great way to incorporate mid-cycle validations, defect fixing and quality audit checkpoints.[1] Although I have reduced the risk of bottlenecks and provided some additional lead time to our critical path, I haven't added any time to my overall project plan.
I also acknowledge that skill level of the resources is not 100% interchangeable. But the fact remains, that if we haven't done this level of workflow analysis, we don't know that we can't redistribute, reorder or restructure to take better use of the resources and skill level that we do have. In this example, resource Orange was required to do Step 3 in WorkflowA, only because there was a portion of Step 3 that needed an advanced level of multi-threading JAVA design. When we take the time to split that piece away from the rest of the component, we find that several other resources could do the rest of Step 3. If we had otherwise identified resource orange had critical skills that no one else had, we could reposition resource Orange into designing and architecting so that others could take his well-designed specifications and easily code from those artifact.
The essence of this tip is that mapping or modeling your workflow early provides multiple options and alternatives. After you are already in the middle of the project and have the programmers entrenched in the code, many of these alternatives are closed to you. You're trapped.

Another advantage of this project management technique is that it avoids the over-padding that we might get when individuals pad for each of their tasks. We place the buffer associated with the critical path that requires the additional time, and not the other tasks. The other tasks (not in the critical path) already have an inherent buffer with the critical long pole.

[1] Use these strategically placed stabilization points to reduce the backlog of defects, documentation reviews and overall house cleaning (See Reduce Inventory Wherever Possible).



Early Critical Path Identification

- Mapping the workflow makes this problem visually clear and provides us a way to avoid it before the project even starts
  - Avoids over-padding by individual task estimations.

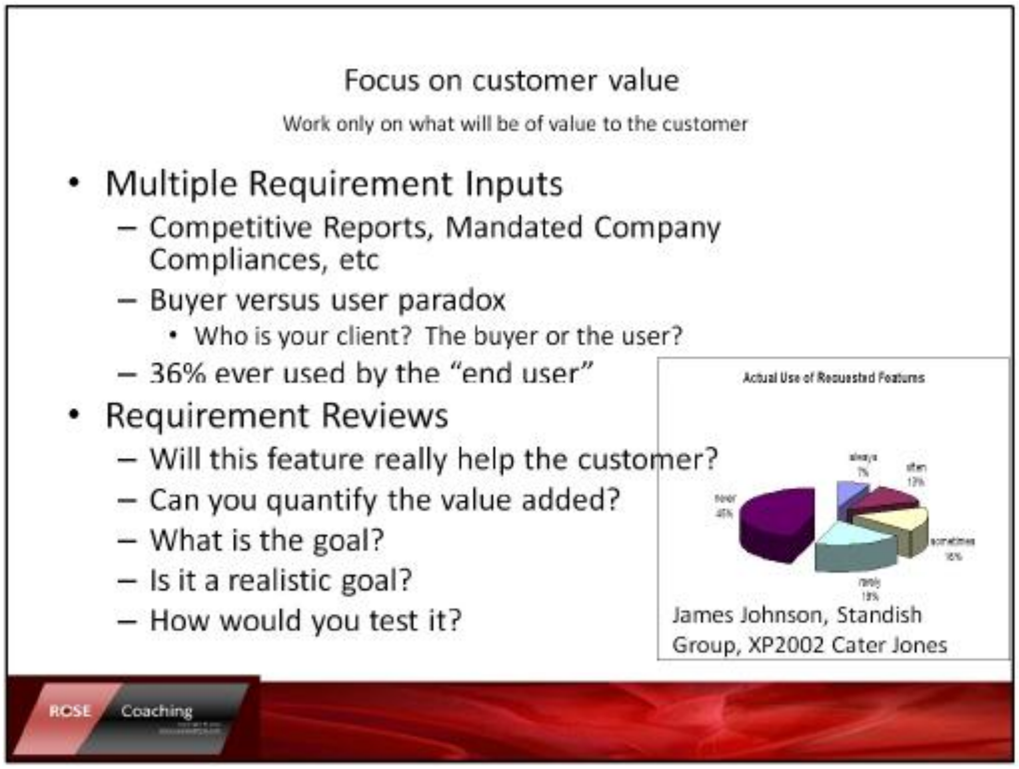
Once you've diagramed your initial flow, you optimize to correct the risks and bottlenecks around the dependent components and resources. In this example, although I've split my tasks into additional steps, my critical path is just 7 days (shorter than my original scheme). I still feel that resource Orange or Step 3a is a potential bottleneck, so I schedule a 2 day buffer before the potential bottleneck. This allows all the sub-steps (Step2a, 2b,2c,2d) to accumulate in a slight holding pattern. This stabilization period is a great way to incorporate mid-cycle validations, defect fixing and quality audit checkpoints.[1] Although I have reduced the risk of bottlenecks and provided some additional lead time to our critical path, I haven't added any time to my overall project plan.

I also acknowledge that skill level of the resources is not 100% interchangeable. But the fact remains, that if we haven't done this level of workflow analysis, we don't know that we can't redistribute, reorder or restructure to take better use of the resources and skill level that we do have. In this example, resource Orange was required to do Step 3 in WorkflowA, only because there was a portion of Step 3 that needed an advanced level of multi-threading JAVA design. When we take the time to split that piece away from the rest of the component, we find that several other resources could do the rest of Step 3. If we had otherwise identified resource orange had critical skills that no one else had, we could reposition resource Orange into designing and architecting so that others could take his well-designed specifications and easily code from those artifact.

The essence of this tip is that mapping or modeling your workflow early provides multiple options and alternatives. After you are already in the middle of the project and have the programmers entrenched in the code, many of these alternatives are closed to you. You're trapped.

Another advantage of this project management technique is that it avoids the over-padding that we might get when individuals pad for each of their tasks. We place the buffer associated with the critical path that requires the additional time, and not the other tasks. The other tasks (not in the critical path) already have an inherent buffer with the critical long pole.

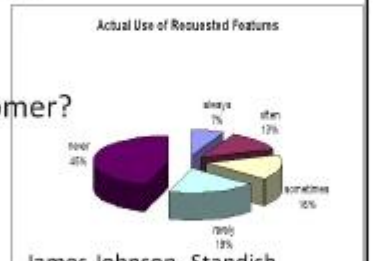
[1] Use these strategically placed stabilization points to reduce the backlog of defects, documentation reviews and overall house cleaning (See Reduce Inventory Wherever Possible).



Focus on customer value

Work only on what will be of value to the customer

- Multiple Requirement Inputs
  - Competitive Reports, Mandated Company Compliances, etc
  - Buyer versus user paradox
    - Who is your client? The buyer or the user?
  - 36% ever used by the "end user"
- Requirement Reviews
  - Will this feature really help the customer?
  - Can you quantify the value added?
  - What is the goal?
  - Is it a realistic goal?
  - How would you test it?



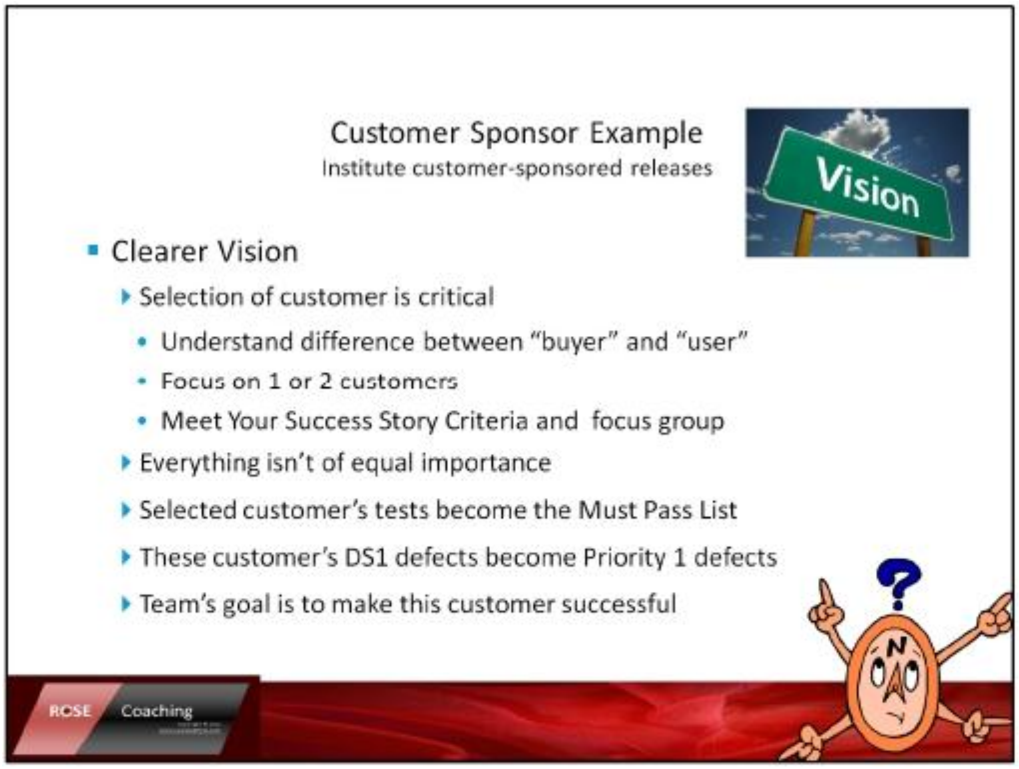
James Johnson, Standish Group, XP2002 Cater Jones

ROSE Coaching

A large study made by James Johnson from the Standish Group (XP 2002) shows that 36% of features of most applications are never used (Figure17). It seems absurd to spend time on things no one will use.

Another way to increase the customer value to your feature list is to have a customer sponsor each release.
In the past, our product managers collect various requirements from various inputs. And they prioritize according to competitive advantage, level of effort and product delivery cycles. In RPT 6.1 – when we did that, and released to several customers, we could not find a success story among those customers. Each customer had their different definition and usage of performance tests, each customer had their particular needs and wants. In the end, we couldn't make everyone happy in the limited time period.
Let's switching our focus "looking for a success story" to "creating the success story".
Product Marketing selects a customer sponsor that illustrates the focus group or "success story" we are trying to accomplish.
At this point, the customer sponsor is involved in the requirement gathering, review and design of this particular release. They demo and evaluate the product at each deliverable iteration. Although we continue to do FVT and SVT testing on all the other features and environments, their test cases and use flows that illustrate how the customer sponsor accomplishes their goals become our Top Must Pass Test Cases. Their obstacles and defects become our High Priority Defects. And their final evaluations become part of our "Go No Go" decisions.
Having customer sponsor releases provide a tangible team goal (make this customer successful) that everyone can understand. Priorities and focus automatically follow. And at the end, we're not looking for a success story, because we've been creating it all along.
The next release, we select a different customer, environment, and customer goal.

Speaker Notes



Customer Sponsor Example

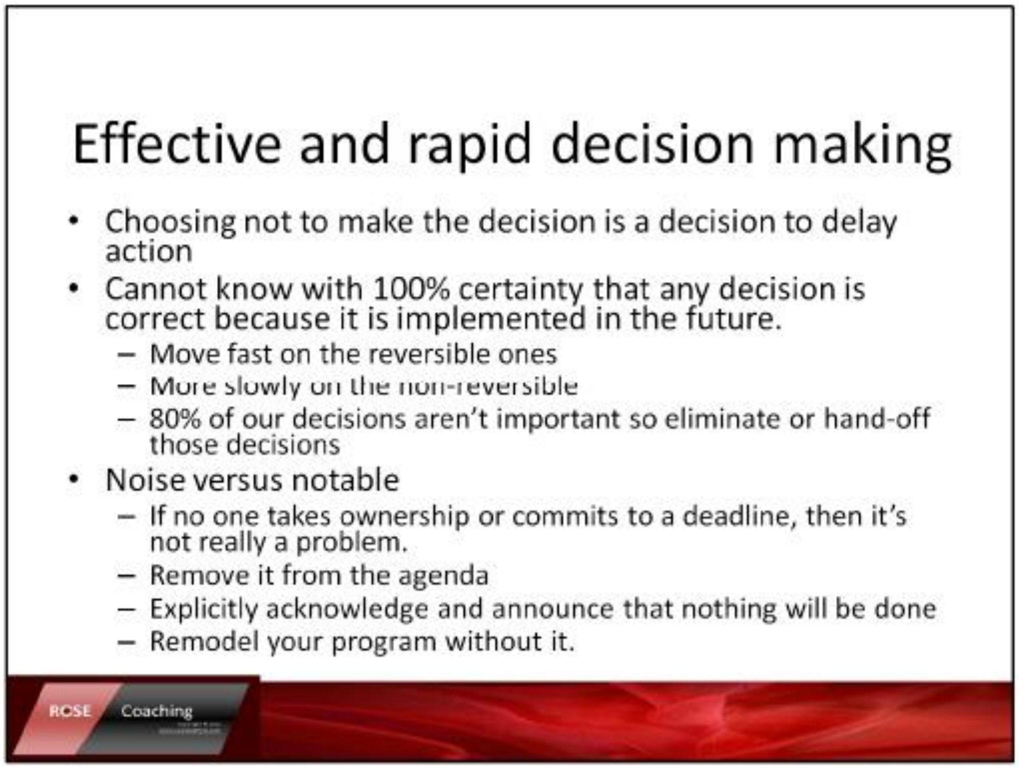
Institute customer-sponsored releases

- Clearer Vision
  - Selection of customer is critical
    - Understand difference between "buyer" and "user"
    - Focus on 1 or 2 customers
    - Meet Your Success Story Criteria and focus group
  - Everything isn't of equal importance
  - Selected customer's tests become the Must Pass List
  - These customer's DS1 defects become Priority 1 defects
  - Team's goal is to make this customer successful

Another way to increase the customer value to your feature list is to have a customer sponsor each release. This is a more formal approach to customer involvement than the customer review suggested in Tip #4.

In the past, our product managers collected various requirements from various inputs, and they would prioritize these according to competitive advantage, level of effort, and the needs of product delivery cycles. Today, product management needs to broaden the reach of this requirements process by selecting a customer sponsor who serves as the focus group for the functionality we are trying to accomplish. The customer sponsor needs to be involved in the requirements gathering, review, and design of this particular release. They evaluate the product at each deliverable iteration. Their test cases and use flows that illustrate how they accomplish their goals become our top "Must Pass" test cases. Their obstacles and defects become our "High Priority" defects. And their final evaluations become part of our "Go / No Go" decisions.

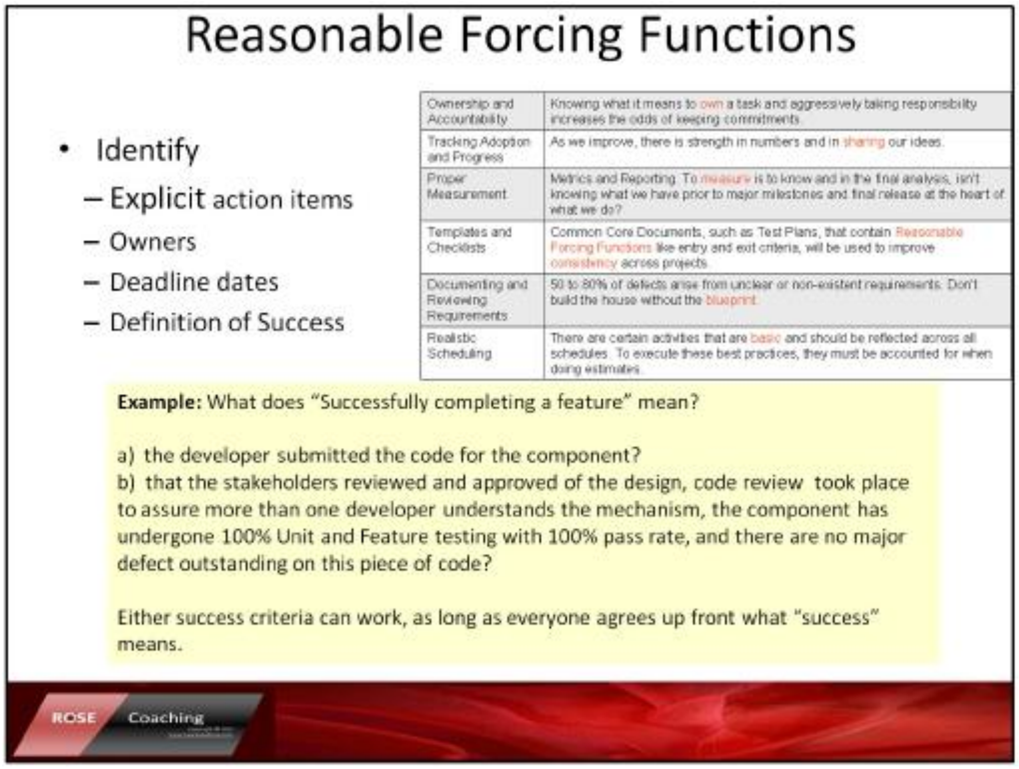
Having customer sponsor releases provide a tangible team goal (make this customer successful) that everyone can understand. Priorities and focus automatically follow. And at the end, we're not looking for a success story, because we've been creating it all along.



Effective and rapid decision making

- Choosing not to make the decision is a decision to delay action
- Cannot know with 100% certainty that any decision is correct because it is implemented in the future.
  - Move fast on the reversible ones
  - More slowly on the non-reversible
  - 80% of our decisions aren't important so eliminate or hand-off those decisions
- Noise versus notable
  - If no one takes ownership or commits to a deadline, then it's not really a problem.
  - Remove it from the agenda
  - Explicitly acknowledge and announce that nothing will be done
  - Remodel your program without it.

One of the hardest collaboration for creating realistic schedules is making decisions.
The reason for delayed in decision making, is often the fear of making the wrong decision. Ironically, choosing not to make the decision is a decision to delay action which doesn't bring you any closer to the right answer. Even the wrong decision brings you closer to a workable solution, because you can immediately deal with the consequences of a "made" decision. The quicker you make the decision, the quicker you can move forward and take action on both the positive and negative results.
I'm not suggesting reckless or snap decision making. I recommend moving fast on the reversible ones, and more slowly on the non-reversible high risk items. If the projected consequences of the wrong choice are minor, make the best decision based on what you know now and move on. In our iterative and constantly changing technical environments, we cannot know with 100% certainty that any decision is correct because it is implemented in the future. So make it and don't worry about it. Once the decision is made, stop discussing it. Execute on it, learn from it and just move on.
On the items you can not decide on today, define the specific action items that will close the gap between where you are today, and where you need to be to make that decision. Make sure you have explicit owners and deadlines for each action (see Rigorously Institute Reasonable Forcing Functions). In our area, we often attend meetings to exhaustively discuss a problem. After much struggle we are dismissed (often because we need to attend another meeting concerning another issue). Therefore, not only have we delayed the decision, we haven't put in place a chain of events to get us any closer. (See Effective Meeting Management and Implement Reasonable Forcing Functions).



Reasonable Forcing Functions

- Identify
  - Explicit action items
  - Owners
  - Deadline dates
  - Definition of Success

Ownership and Accountability	Knowing what it means to own a task and aggressively taking responsibility increases the odds of keeping commitments.
Tracking Adoption and Progress	As we improve, there is strength in numbers and in sharing our ideas.
Proper Measurement	Metrics and Reporting. To measure is to know and in the final analysis, isn't knowing what we have prior to major milestones and final release at the heart of what we do?
Templates and Checklists	Common Core Documents, such as Test Plans, that contain Reasonable Forcing Functions like entry and exit criteria, will be used to improve consistency across projects.
Documenting and Reviewing Requirements	50 to 80% of defects arise from unclear or non-existent requirements. Don't build the house without the blueprint.
Realistic Scheduling	There are certain activities that are basic and should be reflected across all schedules. To execute these best practices, they must be accounted for when doing estimates.

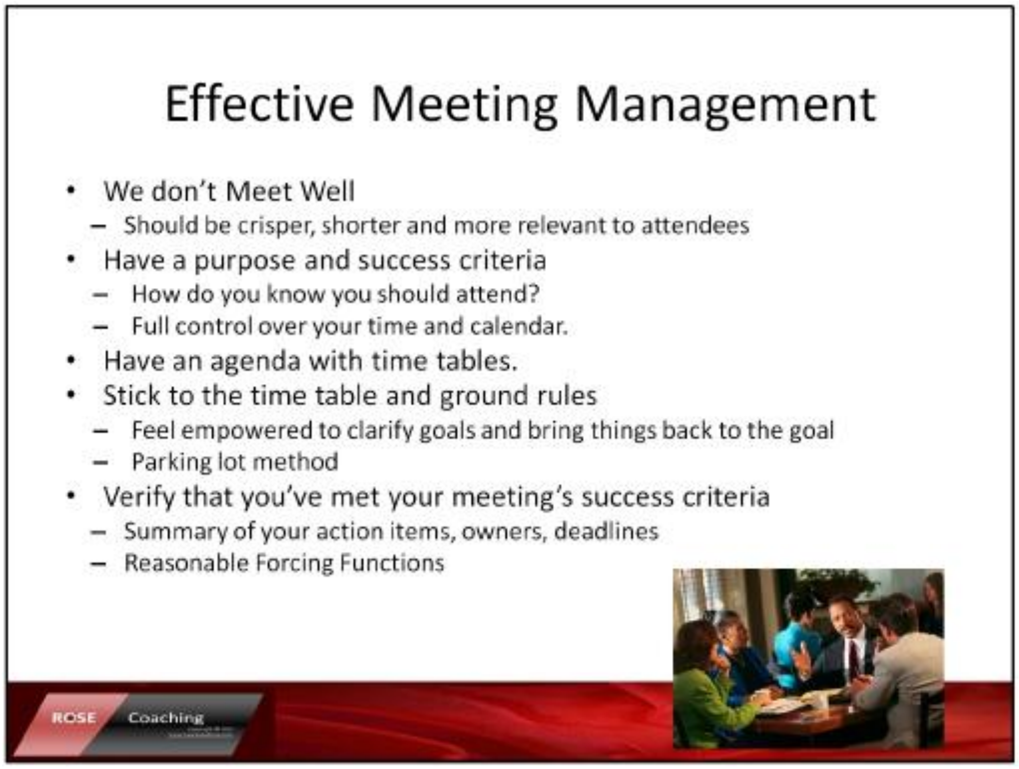
Example: What does "Successfully completing a feature" mean?

a) the developer submitted the code for the component?
b) that the stakeholders reviewed and approved of the design, code review took place to assure more than one developer understands the mechanism, the component has undergone 100% Unit and Feature testing with 100% pass rate, and there are no major defect outstanding on this piece of code?

Either success criteria can work, as long as everyone agrees up front what "success" means.

ROSE Coaching

As you can see from the previous tips, I've mention identifying explicit action items, tasks, owners and deadline dates. This is essentially what Reasonable Forcing Functions are all about. To assure success in any endeavor, you need not only a successful plan, but explicit, time boxed task lists with responsible, accountable owners and acknowledged consequences
Other examples are defining Success Criteria for various actions or programs.
Steven Covey would describe this as "Starting with the End in Mind".
If an activity is to succeed, we need to understand the definition of success.



Effective Meeting Management

- We don't Meet Well
  - Should be crisper, shorter and more relevant to attendees
- Have a purpose and success criteria
  - How do you know you should attend?
  - Full control over your time and calendar.
- Have an agenda with time tables.
- Stick to the time table and ground rules
  - Feel empowered to clarify goals and bring things back to the goal
  - Parking lot method
- Verify that you've met your meeting's success criteria
  - Summary of your action items, owners, deadlines
  - Reasonable Forcing Functions

ROSE Coaching

As described in the "paradox of busy", the busier we are, the more we find ourselves communicating and working in teams. This often means more meetings. Since we spend a majority of our communication time in meetings, they need to be effective, purposeful and meaningful.

There is much written on Effective Meeting Management, which I don't intend to cover. We all understand the jest:

Have a purpose and success criteria for this meeting

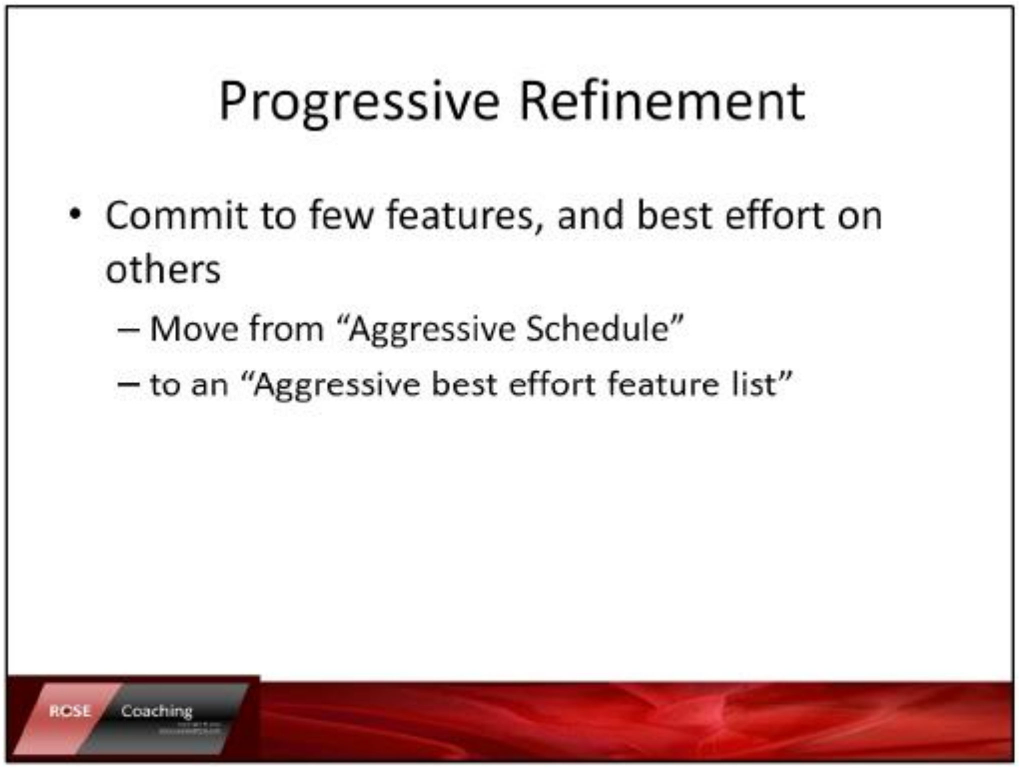
Have an agenda with time tables.

Stick to your time table and meeting ground rules

Don't leave without verifying that you've met your meeting's success criteria with a Summary of your action items, owners, deadlines (See Rigorously Institute Reasonable Forcing Functions)

We all know the attributes of a successful meeting; we just need to execute it successfully.

Controversial Litmus Test: If you felt you just came out of an ineffective meeting, it's your fault. It doesn't matter whether you facilitated it or just participated in it. You are ultimately responsible for the way you spend your time. If you attended a meeting without understanding the purpose, agenda, or reasonable forcing success criteria, shame on you.



Progressive Refinement

- Commit to few features, and best effort on others
  - Move from "Aggressive Schedule"
  - to an "Aggressive best effort feature list"

ROSE Coaching

We've all heard the groan that "we have an aggressive schedule to meet". The fact is that the "schedule" timeline isn't aggressive. It's what we have chosen to fit in it[1]. We acknowledged earlier that we cannot know with 100% certainty that any decision is correct because it's implemented in the future. By accepting that time changes our environment and sometimes even our purpose; we also need to accept progressive refinement. The concept is to define a very conservative stance in the early inception of the project schedule. Commit to only a conservative few (3-5) features and provide a "best effort" on the rest of the feature sets. And as time progresses, and we learn more, we refine our estimates and schedules accordingly. This allows us to change our energies from the complaint of "an aggressive schedule", to a realistic but aggressive feature list.

Progressive refinement isn't delaying a decision on the schedule. We're actually making a series of decisions, consistently reviewing and refining throughout the project.

I am also not condoning excess padding. Instead, incorporate a reasonable amount of stabilization periods between critical paths and bottlenecks (See Identify Critical Paths and Bottlenecks early), and institutes "confidence percentage" as part of your schedule assessment and review criteria

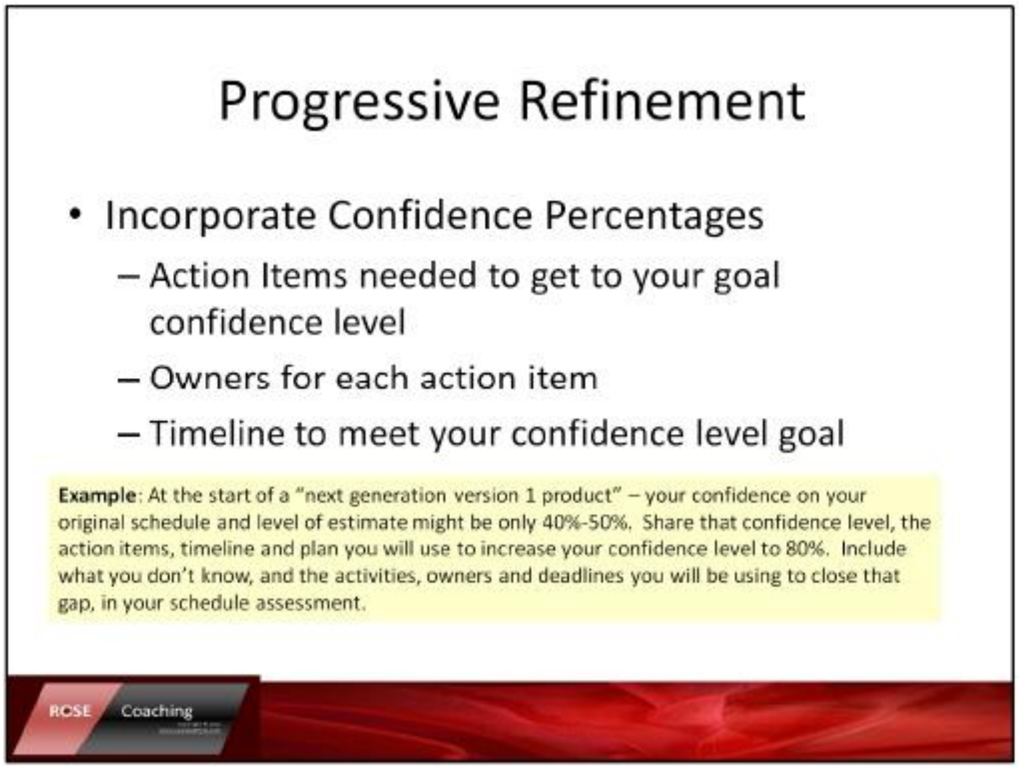
Often times we feel pressure to sign up to a schedule and commit[2], too early in the development phase. The schedule might look "fine" for what you know today, so you can't exactly state you disagree with it. It's just that you really "don't know". Incorporating the "confidence percentage" strategy allows you to comfortably agree and also illustrates your areas of concern.

For instance, at the start of a "next generation version 1 product" – your confidence on your original schedule and level of estimate might be only 40%-50%. Share that confidence level, the action items, timeline and plan you will use to increase you confidence level to 80%. Include what you don't know, and the activities, owners and deadlines you will be using to close that gap, in your schedule assessment.

[1] For instance, if we try to put 10 pounds of stuff in a 5 pound bag – it is the bag's fault?

[2] Many high level project managers like to ceremonially go around the room getting verbal commitment on early skeleton schedule. The intent is to get the individual practitioners to verbally commit and be held accountable for the work. Even though the intent is a good one, the timing may not be appropriate. We may not know and understand enough about the program at this point, to comfortable "commit" to the schedule at this point.

Speaker Notes



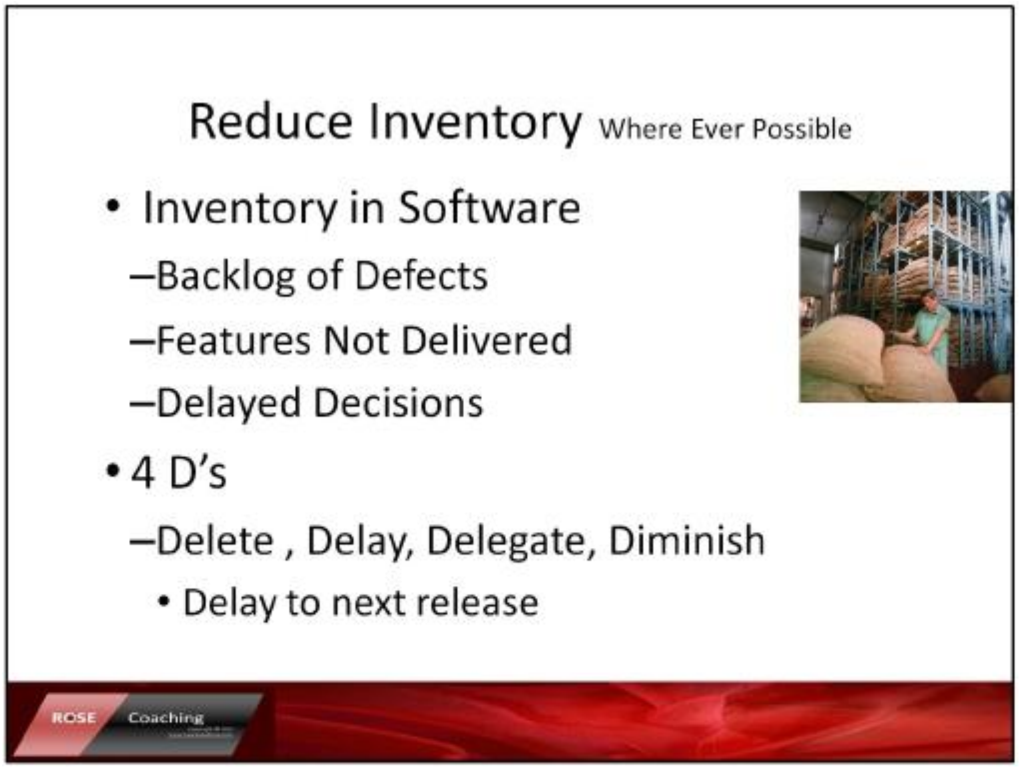
Progressive Refinement

- Incorporate Confidence Percentages
  - Action Items needed to get to your goal confidence level
  - Owners for each action item
  - Timeline to meet your confidence level goal

Example: At the start of a "next generation version 1 product" – your confidence on your original schedule and level of estimate might be only 40%-50%. Share that confidence level, the action items, timeline and plan you will use to increase your confidence level to 80%. Include what you don't know, and the activities, owners and deadlines you will be using to close that gap, in your schedule assessment.

We've all heard the groan that "we have an aggressive schedule to meet". The fact is that the "schedule" timeline isn't aggressive. It's what we have chosen to fit in it[1]. We acknowledged earlier that we cannot know with 100% certainty that any decision is correct because it's implemented in the future. By accepting that time changes our environment and sometimes even our purpose; we also need to accept progressive refinement. The concept is to define a very conservative stance in the early inception of the project schedule. Commit to only a conservative few (3-5) features and provide a "best effort" on the rest of the feature sets. And as time progresses, and we learn more, we refine our estimates and schedules accordingly. This allows us to change our energies from the complaint of "an aggressive schedule", to a realistic but aggressive feature list.
Progressive refinement isn't delaying a decision on the schedule. We're actually making a series of decisions, consistently reviewing and refining throughout the project.
I am also not condoning excess padding. Instead, incorporate a reasonable amount of stabilization periods between critical paths and bottlenecks (See Identify Critical Paths and Bottlenecks early), and institutes "confidence percentage" as part of your schedule assessment and review criteria
Often times we feel pressure to sign up to a schedule and commit[2], too early in the development phase. The schedule might look "fine" for what you know today, so you can't exactly state you disagree with it. It's just that you really "don't know". Incorporating the "confidence percentage" strategy allows you to comfortably agree and also illustrates your areas of concern.
For instance, at the start of a "next generation version 1 product" – your confidence on your original schedule and level of estimate might be only 40%-50%. Share that confidence level, the action items, timeline and plan you will use to increase you confidence level to 80%. Include what you don't know, and the activities, owners and deadlines you will be using to close that gap, in your schedule assessment.

[1] For instance, if we try to put 10 pounds of stuff in a 5 pound bag – it is the bag's fault?
[2] Many high level project managers like to ceremonially go around the room getting verbal commitment on early skeleton schedule. The intent is to get the individual practitioners to verbally commit and be held accountable for the work. Even though the intent is a good one, the timing may not be appropriate. We may not know and understand enough about the program at this point, to comfortable "commit" to the schedule at this point.

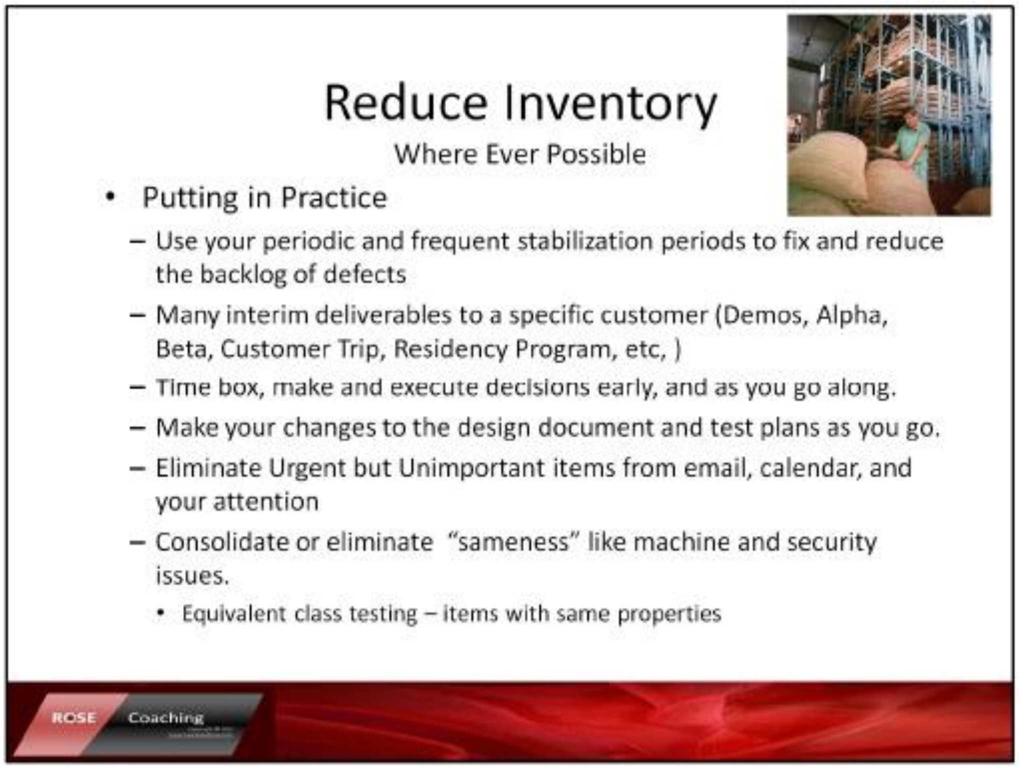


Reduce Inventory Where Ever Possible

- Inventory in Software
  - –Backlog of Defects
  - –Features Not Delivered
  - –Delayed Decisions
- 4 D’s
  - –Delete , Delay, Delegate, Diminish
    - Delay to next release

ROSE Coaching

The retail industry like clothing stores, grocers, hardware shops, and others understand the cost of keeping a large inventory. Their goal is to stock the shelves just enough to keep the merchandize moving. Large inventories means lost of money, because the dress might go out of style, the food might spoil, the mechanize might not sell and you’re stuck with the entire backlog.
In the software industry the concept is similar, except our inventory consists of a backlog of defects, a list of features that aren’t ready and a series of unmade decisions. The wait cycle on these items are costly because technology (and customers) is always changing. Ways to reduce these wait times are:



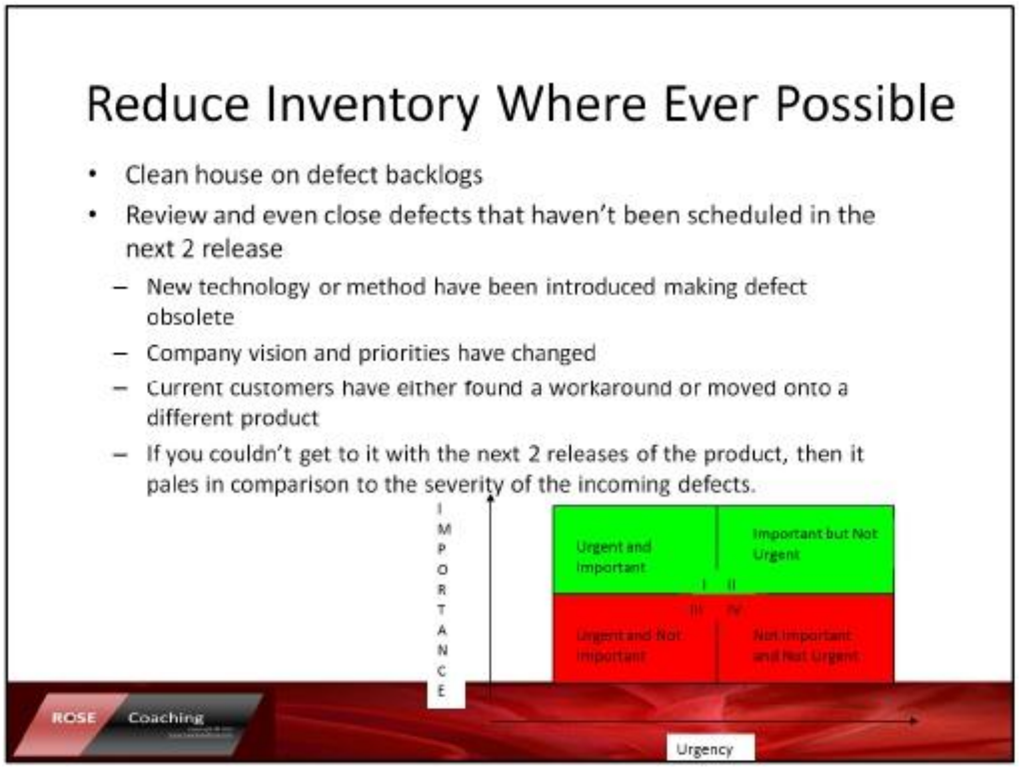
Reduce Inventory

Where Ever Possible

- Putting in Practice
  - Use your periodic and frequent stabilization periods to fix and reduce the backlog of defects
  - Many interim deliverables to a specific customer (Demos, Alpha, Beta, Customer Trip, Residency Program, etc,)
  - Time box, make and execute decisions early, and as you go along.
  - Make your changes to the design document and test plans as you go.
  - Eliminate Urgent but Unimportant items from email, calendar, and your attention
  - Consolidate or eliminate "sameness" like machine and security issues.
    - Equivalent class testing – items with same properties

ROSE Coaching

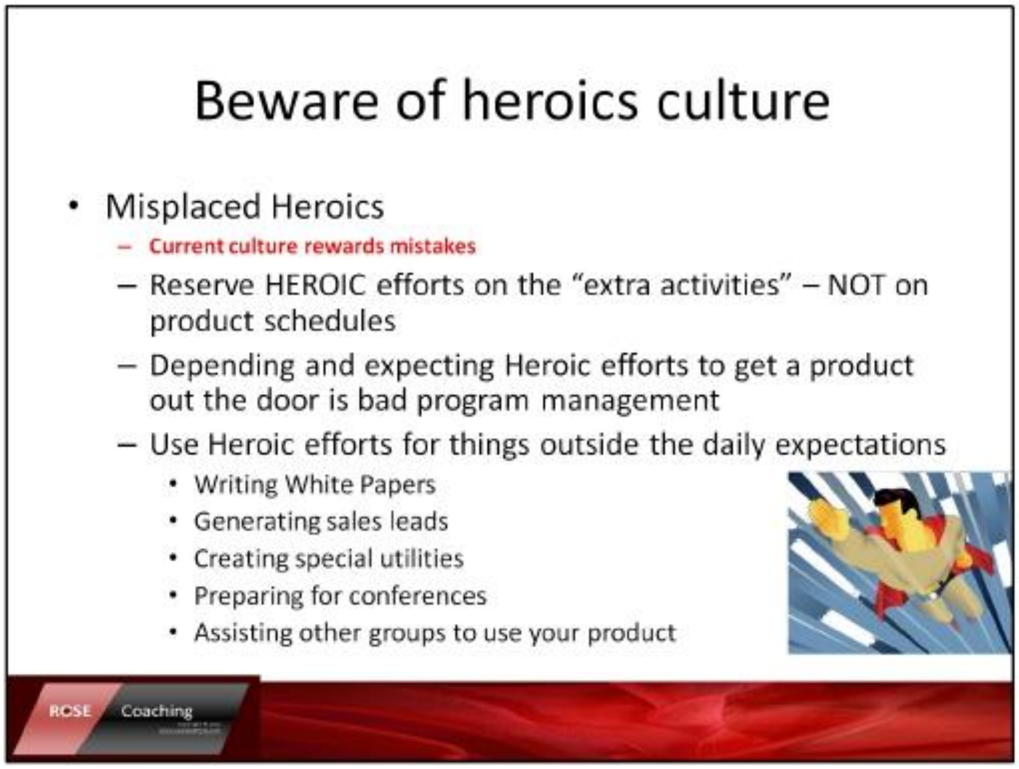
The retail industry like clothing stores, grocers, hardware shops, and others understand the cost of keeping a large inventory. Their goal is to stock the shelves just enough to keep the merchandize moving. Large inventories means lost of money, because the dress might go out of style, the food might spoil, the mechanize might not sell and you're stuck with the entire backlog.
In the software industry the concept is similar, except our inventory consists of a backlog of defects, a list of features that aren't ready and a series of unmade decisions. The wait cycle on these items are costly because technology (and customers) is always changing.
Ways to reduce these wait times are:



Reduce Inventory Where Ever Possible

- Clean house on defect backlogs
- Review and even close defects that haven't been scheduled in the next 2 release
  - New technology or method have been introduced making defect obsolete
  - Company vision and priorities have changed
  - Current customers have either found a workaround or moved onto a different product
  - If you couldn't get to it with the next 2 releases of the product, then it pales in comparison to the severity of the incoming defects.

- Exercise: Take the list you created at the start of this workbook. Critically review and identify which Quadrant they actually fit in. If you are spending time on Q3, devise a plan to remove or reduce that time – to create more Q2 time.



Beware of heroics culture

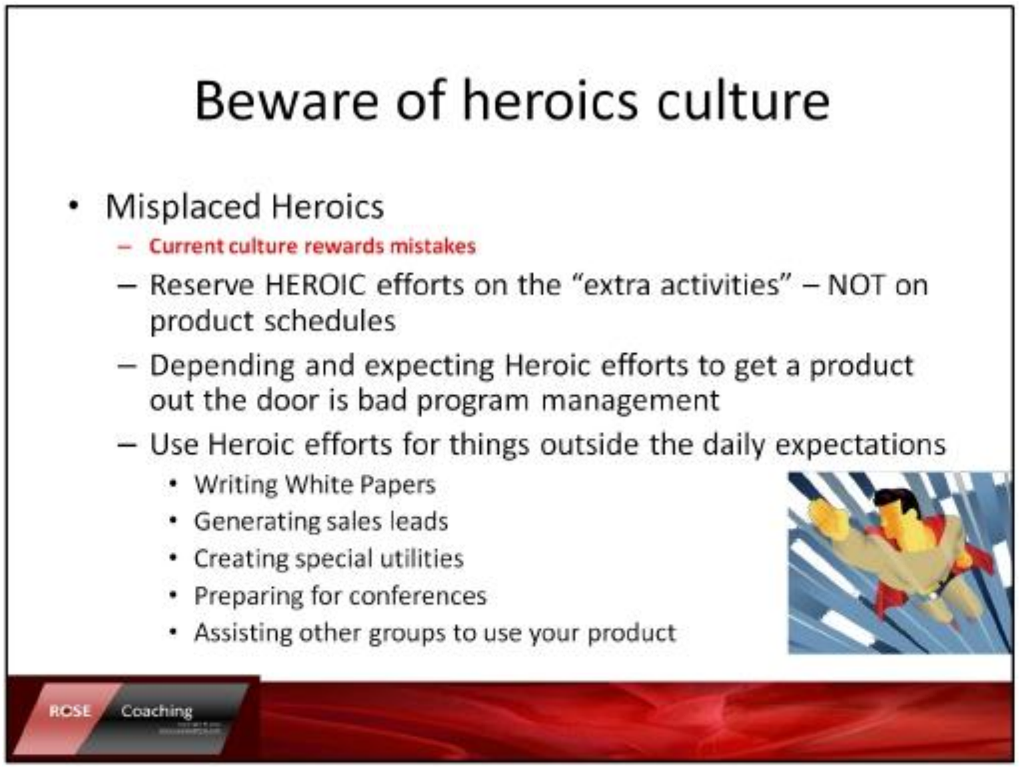
- Misplaced Heroics
  - Current culture rewards mistakes
  - Reserve HEROIC efforts on the "extra activities" – NOT on product schedules
  - Depending and expecting Heroic efforts to get a product out the door is bad program management
  - Use Heroic efforts for things outside the daily expectations
    - Writing White Papers
    - Generating sales leads
    - Creating special utilities
    - Preparing for conferences
    - Assisting other groups to use your product

ROSE Coaching

Misplaced placed Heroics is extremely common and another source of chaos.

Another form of chaos is misplaced heroics. It is extremely common that organizations depend and expect additional hours and heroics to get their products out the door. While very common, it doesn't make it good management. Depending upon heroics to get your product out the door is simply bad program management. If you built a realistic schedule in the first place, then heroics would not be required. Failing to properly identify the risks and delays, is bad program management. Assigning a single resources on parallel projects without the proper buffers and supporting cast is bad program management. The Capability Maturity Model states it simply as a Level 1 mentality.
Save the heroics for the additional tasks outside of the daily needs of releasing a customer value product. Some examples are white papers on successful deployment of the products, papers on hints and tips for the customer, presenting at conferences, etc.

Speaker Notes



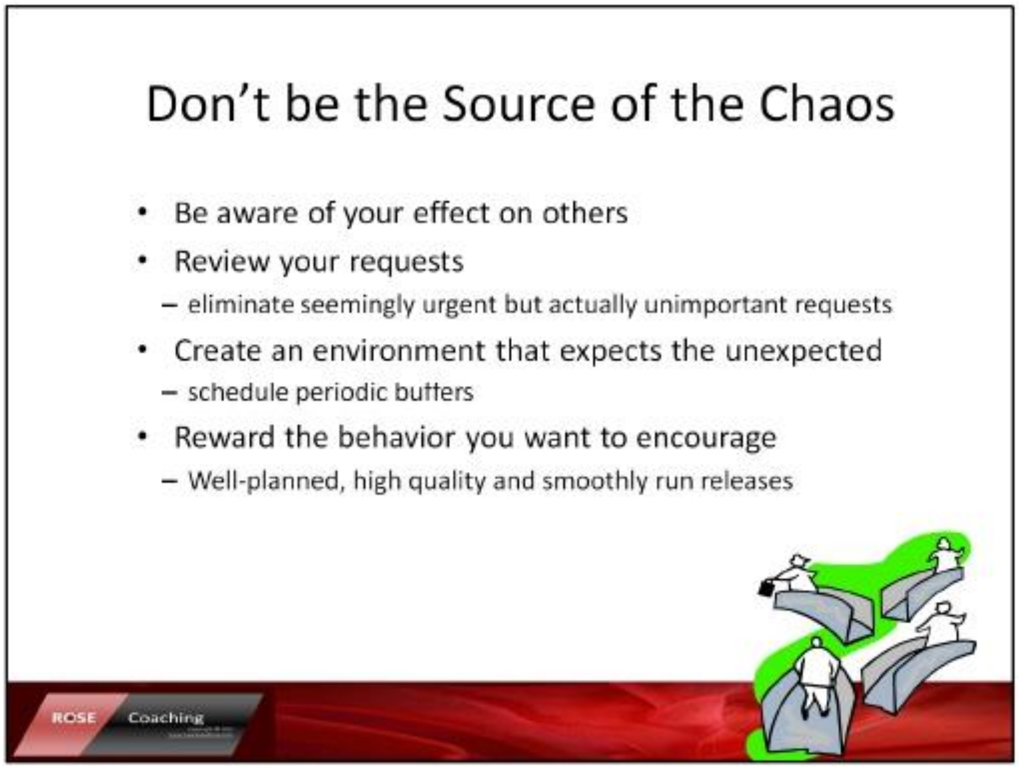
Beware of heroics culture

- Misplaced Heroics
  - Current culture rewards mistakes
  - Reserve HEROIC efforts on the "extra activities" – NOT on product schedules
  - Depending and expecting Heroic efforts to get a product out the door is bad program management
  - Use Heroic efforts for things outside the daily expectations
    - Writing White Papers
    - Generating sales leads
    - Creating special utilities
    - Preparing for conferences
    - Assisting other groups to use your product

ROSE Coaching

Misplaced placed Heroics is extremely common and another source of chaos.

Another form of chaos is misplaced heroics. It is extremely common that organizations depend and expect additional hours and heroics to get their products out the door. While very common, it doesn't make it good management. Depending upon heroics to get your product out the door is simply bad program management. If you built a realistic schedule in the first place, then heroics would not be required. Failing to properly identify the risks and delays, is bad program management. Assigning a single resources on parallel projects without the proper buffers and supporting cast is bad program management. The Capability Maturity Model states it simply as a Level 1 mentality.
Save the heroics for the additional tasks outside of the daily needs of releasing a customer value product. Some examples are white papers on successful deployment of the products, papers on hints and tips for the customer, presenting at conferences, etc.

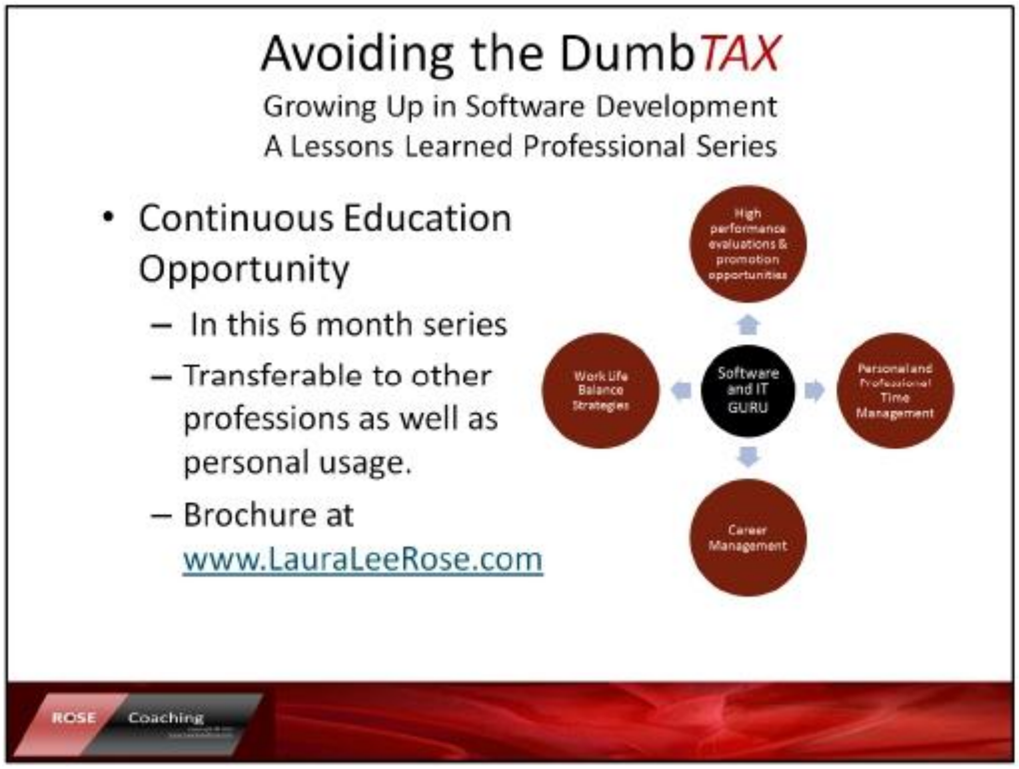


Don't be the Source of the Chaos

- Be aware of your effect on others
- Review your requests
  - eliminate seemingly urgent but actually unimportant requests
- Create an environment that expects the unexpected
  - schedule periodic buffers
- Reward the behavior you want to encourage
  - Well-planned, high quality and smoothly run releases

ROSE Coaching

As a leader, don't be the source of the chaos. You're schedule may hectic and unorganized, but being a good leader means not having your mis-managed time affect other members of your team.
Examples:
· If you have an emergency come up, do not move up a meeting because you now have to fly out of town. Instead, delegate and train someone else to lead that meeting.
· If you misjudged how long a project will take, do not make your team work 24-7 to make up for your mistake. Instead, conduct a formal change management meeting to create new expectations and inform all stakeholders.
· If you feel your personality styles clash with a client, do not allow your ego to sour the overall client experience. Instead, select someone with a more matching personality as liaison to this client.



Avoiding the DumbTAX

Growing Up in Software Development
A Lessons Learned Professional Series

- Continuous Education Opportunity
  - In this 6 month series
  - Transferable to other professions as well as personal usage.
  - Brochure at www.LauraLeeRose.com

High performance evaluations & promotion opportunities
Work Life Balance Strategies
Software and IT GURU
Personal and Professional Time Management
Career Management
ROSE Coaching

Continuing Education Opportunities include more information on Critical Path Analsys, Change Management, Risk Management and Analsys, Team Organization and Management and much more.

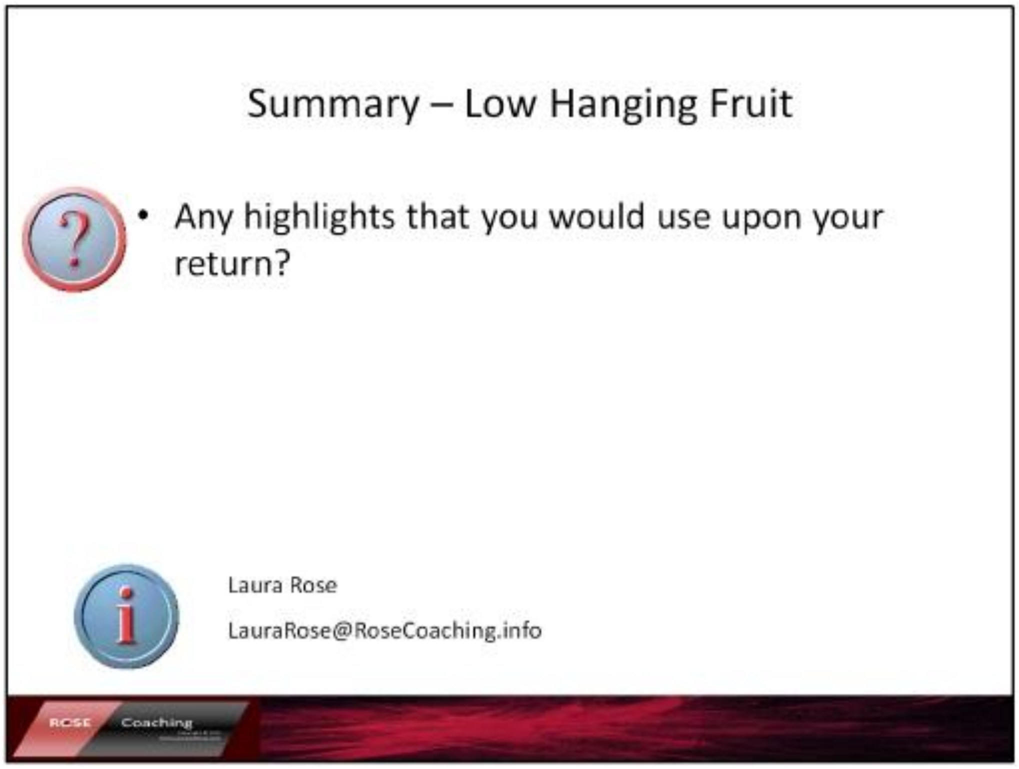
Classes can be registered individually, 3 month or 6 month packets.
Class brochure: www.LauraLeeRose.com

Register at: http://lessonslearnedseries.eventbrite.com

More Time Management tips can be found in my book TimePeace: Making peace with time (which can be found at www.amazon.com)

Access to free articles can be found by subscribing to the library at http://eepurl.com/dUi81

- Exercise: Do an objective skill-gap analysis of yourself. Identify which areas you need improvement. Review the articles and additional resources for Continuous Educational Opportunities.



Summary – Low Hanging Fruit

- Any highlights that you would use upon your return?

Laura Rose

LauraRose@RoseCoaching.info

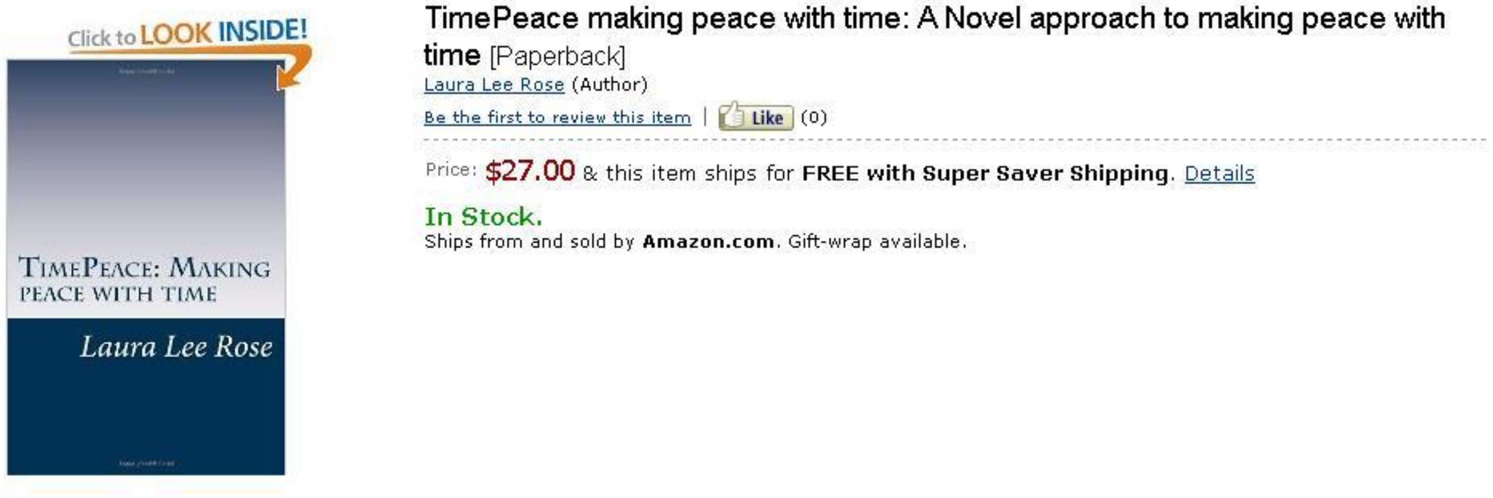
ROSE Coaching

Transforming the life you have into the life you really want.

ASSISTING CORPORATE PROFESSIONALS IN THEIR TRANSITION INTO THAT NEXT POSITION, A NEW CAREER OR THEIR OWN BUSINESS.

Do you need more time to:

- Learn a new skill
- Start a side business
- Exercise
- Travel
- Follow-through on goals, commitments and resolutions
- Volunteer
- Read and self-study
- Have fun with hobbies, friends and family



TimePeace making peace with time: A Novel approach to making peace with time [Paperback]

Laura Lee Rose (Author)

Be the first to review this item | Like (0)

Price: $27.00 & this item ships for FREE with Super Saver Shipping. Details

In Stock.

Ships from and sold by Amazon.com. Gift-wrap available.

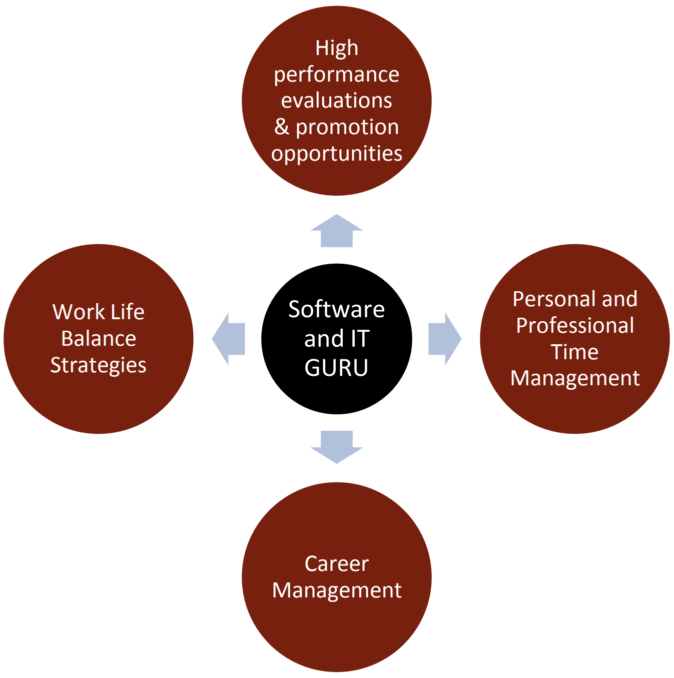
GoTo Academy: Tools for the GoTo Guy and Gal

Growing Up in Software Development

A Lessons Learned Professional Series

In this 6 month series (weekly series), we will cover 4 critical categories of our personal and professional lives. Even though the scenarios will be 'IT specific', the concepts are transferable to other professions as well as personal usage. You will be send learning modules designed in 5-10 minute chunks.

More info: Find out more


High performance evaluations & promotion opportunities
Work Life Balance Strategies
Software and IT GURU
Personal and Professional Time Management
Career Management

Register Now!

Detail curriculum and tentative schedule

Personal and Professional Time Management	
12 Tips to Realistic Scheduling	Week 1
Simple Techniques to Change Management and Critical Path Analysis	Week 7
Tips for IT Professionals to Increase Effectiveness and Efficiency in the office	Week 8
How to stay focused when working from home	Week 8
Secrets behind accurate estimating	Week 2
How to handle multiple and unclear job directives	Week 5
Finding time to learn new skills	Week 6
10 Career Boosting Resolutions	Week 3
Professional handling delays in your projects	Week 5
Career Management	
Career Maintenance 101	Week 3
How to Say YES on your own terms	Week 9
How to transition into consulting OR back from consulting	Week 10
Staying Visible as a remote employee or manager	Week 8
Thinking like the CEO	Week 10
Finding time to network (and its importance)	Week 11
Recognizing promotion opportunities	Week 10
High performance evaluations and promotion opportunities	
The secrets behind on-time delivery of quality products	Week 1
Effective Client Engagements lead to Superior Client Experiences	Week 12
Understanding the Client Perspective on Defects and Service Issues	Week 12
Building Your Entourage - Creating supporters and sponsors	Week 10
Finding time to learn new skills	Week 6
10 Career Boosting Resolutions	Week 3
Professional handling delays in your projects	Week 1
Recognizing promotion opportunities	Week 10
Work Life Balance Strategies	
Finding FUN in everything that you do	Week 12
Finding time to learn new skills	Week 6
10 Career Boosting Resolutions	Week 3
Building Your Entourage - Creating supporters and sponsors	Week 10
How to Say YES on your own terms	Week 9
Volunteering your way to your next career	Week 11
Finding time to network (and its importance)	Week 11
Recognizing promotion opportunities	Week 10

Register Now!

Notes: