Catalysts

Agile Meets Theory of Constraints

Christoph Steindl

steindl@catalysts.cc

Agenda

Catalysts

- What is the Theory of Constraints?
  - Cost versus Throughput
  - The 5 Focusing Steps
  - The 5 Tools
  - TOC Summary
- How does the TOC relate to us
  - How do we identify constraints?
  - How do we handle constraints?

"The detail is meaningless
until you understand the paradigm."

Michael Kennedy

What is the Theory of Constraints?

Catalysts

- The Theory of Constraints views systems as chains. The weakest link is the constraint that keeps the chain from doing any better at achieving its goal.

- The Theory of Constraints is essentially about change:
  - What to change? (Where is the constraint?)
  - What to change to? (What should we **do** with the constraint?)
  - How to cause the change? (How do we implement the change?)

- It is a system philosophy.
  - It poses these questions at the system-level.
  - It requires that at least someone knows what the ultimate goal of the system is.
  - It then helps to establish the necessary conditions for reaching the goal.

Cost versus Throughput

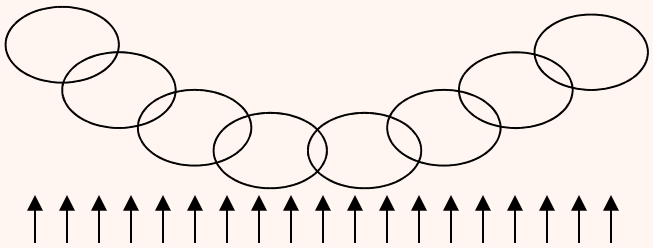
Catalysts

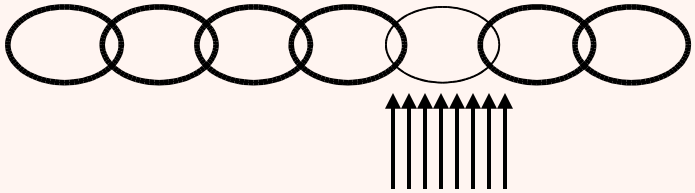
Cost World

- Optimize the cost of development
- Optimize the weight of each link of the chain
- Improvement on any one link improves the wheight of the entire chain
- Local improvements are rewarding

Throughput World

- Optimize the throughput of development
- Optimize the strength of each link of the chain
- Only on the weakest link does improvement strengthen the entire chain
- Local improvements are only rewarding on the bottleneck





The 5 Focusing Steps

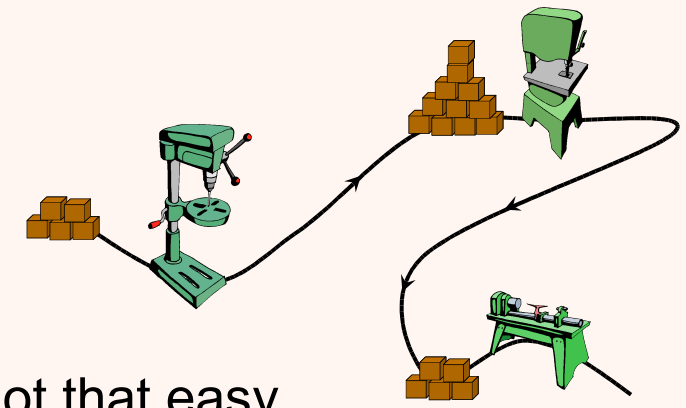
Catalysts

1. Identify the System Constraint
2. Decide How to Exploit the Constraint
3. Subordinate Everything Else
4. Elevate the Constraint
5. Go Back to Step 1, but Beware of “Inertia”

- No effort on non-constraints will produce immediate, measurable improvement in system capability.
- Measure improvement at the system level.
  - E.g. average cycle time, throughput, net profit, customer satisfaction
  - Local measurements only lead to sub-optimization

Works Well in Manufacturing

Catalysts



- It is often easy to spot the bottleneck
  - If the bottleneck is a physical constraint.
  - Where does the inventory pile up?

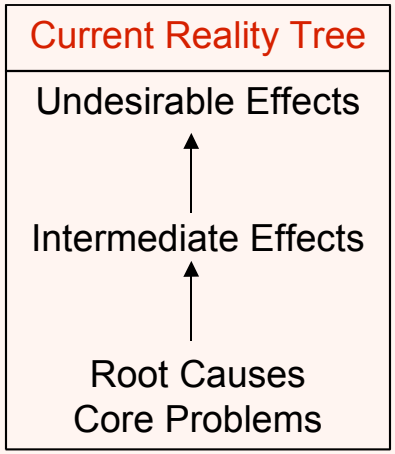
- But in software engineering, it is often not that easy
  - Here, physical constraints are not so common.
  - Typically we are constrained by policies
    - Rules, plans, procedures, measurements, or other guidance
    - "We've always done it that way."
  - Policy constraints are usually much more devastating than physical constraints.
  - Nearly every physical constraint results from some policy constraint.

- We need tools to identify and eliminate the policy constraints

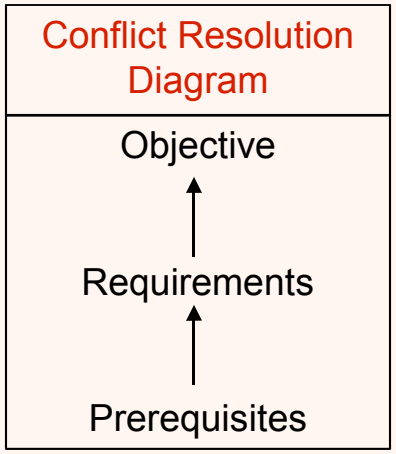
The 5 Tools

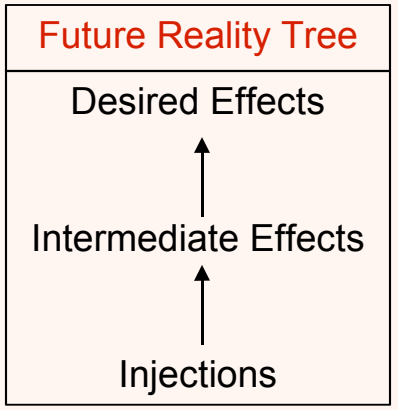
Catalysts

WHAT to change?

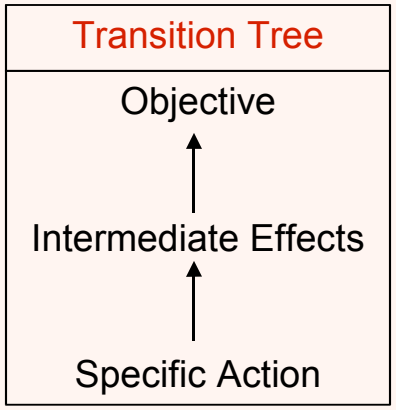


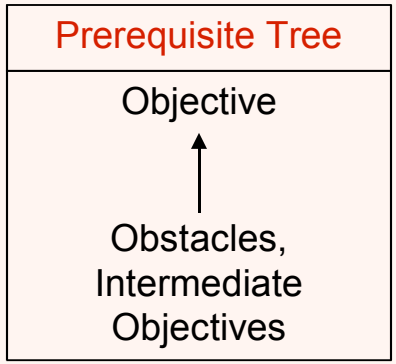
What to change TO?





How to CAUSE the change?





The 5 Tools

Catalysts

- The **Current Reality Tree** – identify the problem
  - Examines the cause-effect logic behind the current situation
  - Identifies the core problem constraining the system
- The **Conflict Resolution Diagram** – create an idea
  - Resolves hidden conflicts that usually perpetuate chronic problems
  - Generates ideas for new, “breakthrough” solutions
- The **Future Reality Tree** – test the idea
  - Verifies that an action will produce the desired results
  - Identifies any unfavorable new consequences
- The **Prerequisite Tree** – overcome obstacles
  - Identifies the obstacles and the best ways to overcome those obstacles for implementing the decision
  - Sequences the major milestones
- The **Transition Tree** – develop step-by-step execution instructions
  - Details the step-by-step instructions for implementing a course of action
  - Provides the rationale for each step

TOC Summary - Agile

Catalysts

- The Theory of Constraints
  - is based on the chain analogy (weakest link)
  - is essentially about change, i.e. improvement
  - keeps us from sub-optimizing by measuring success at the system level
  - gives us tools for analytical thinking (i.e. analyzing a situation, for generating ideas for improvement, for testing them and for planning their implementation)
- Isn't it too obvious to be of value?

- Agile Methodologies
  - are based on the agile manifesto
  - Are essentially about delivering working software
  - Keep us from sub-optimizing by measuring customer satisfaction
  - Give us values, principles and practices for delivering working software
- Scrum: "It's about common sense"

TOC Summary – TQM / CPI

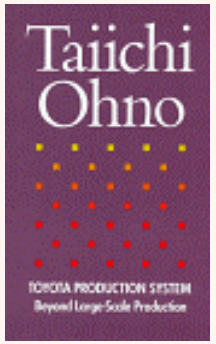
Catalysts

- The Theory of Constraints
  - is essentially about improvement of goal achievement
  - involves only a few, the parts related to the constraint ➔ improvements are incremental and attributable to specific projects or activities
  - works well in manufacturing and also in more complex, less structured areas
  - zeroes in on the system's constraint – the critical one, as opposed to the trivial many
  - needs little capital investment and often has big short-term returns
- Total Quality Management (TQM) and Continuous Process Improvement (CPI)
  - are essentially about process improvement
  - involve large parts (or everyone) and strives to improve all aspects of the system at the same time ➔ tracing back improvements to actions becomes impossible
  - work well in tweaking manufac-turing processes to greater efficiency (suboptimzation?)
  - strengthen all links; often that makes the chain heavier than it needs to be, i.e. more costly!
  - need a sustained effort, a lot of money and often has small returns even in the long run

TOC and Lean Thinking

Catalysts

- From “The Toyota Production System” by Taichii Ohno
  - Eliminate Waste
  - Expose Problems – Stop the line



- From “Lean Software Development” by Mary and Tom Poppendieck
  - Engage Workers’ Intelligence
  - Optimize the Whole
    - With holistic measurements (average cycle time, business case goals, customer satisfaction)

How does the TOC relate to us?

Catalysts

- Scrum engages everyone in identifying obstacles. TOC ties up to that by generating ways to overcome the obstacles and by laying out a (verified) plan for implementation.
- TOC takes people's feelings and the organization's culture seriously when identifying obstacles. Change is never painless, but a judicious application of the TOC tools can dramatically smooth the transition and reduce discomfort.
- TOC only works if you have a thorough knowledge of the subject, just like Scrum relies on the team to have all relevant knowledge.
- TOC gives you thinking tools, whereas Scrum just relies on emergence and self-organization.
- TOC increases our confidence in proposed changes (we can play through the likely consequences).

How do we identify constraints?

Catalysts

- During the planning meeting
  - identification of constraints: what fits in the current iteration, what must be postponed
  - synchronization of the entire team (development + customer) on vision, iteration goals and progress
- During the Daily Scrum meeting
  - identification of constraints, lacks of knowledge, etc.
  - team synchronization for exploiting and subordination (e.g. help each other, jump in, relieve work from the critical resource)
  - ScrumMaster to know the obstacles and work on elevating them
- Via Information Radiators
  - identification of constraints when passing by / “on the fly”
  - shared understanding and buy-in from whole team
- During the Sprint Review / Retrospectives
  - team learning
  - shared understanding and buy-in

Known Constraints and the Agile Answers

Catalysts

- If something troubles you, **think** about it.
  - Or even better: get a group of people to think about it
  - “Engage the workers’ intelligence” (principle from LSD)
- If something is difficult
  - Increase the **feedback** (get more feedback)
  - Get feedback more often
- Examples:
  - Late Integration ➔ Continuous Integration
  - Late Testing ➔ Test-Driven Development / Test-First Design
  - Late Bug Detection ➔ Pair Programming
  - Truck-Factor of 1 ➔ Collective Ownership of Code, Whole Team, Osmotic Communication
  - Cost curve for late changes ➔ Refactoring
  - Regression Errors ➔ Little Change Detectors (Unit Tests)

How do we handle constraints?

Catalysts

- We rely on the creativity and self-organization of the team.
  - The Product Owner and ScrumMaster handle external constraints.
  - The team handles internal constraints.
- This approach works well, it engages the people with the profound knowledge of the system.
- Agile teams are typically diverse teams of *generalizing specialists*[1], for good reason:
  - Often, people are the bottleneck, the "Capacity Constrained Resources."
  - We try to reduce the effects of the bottlenecks by having holistic teams of generalizing specialists.
  - We have more points of view in the group brainstorming sessions.

- But, this approach can benefit from thinking tools as those from the Theory of Constraints.
- They give us a straightforward way to tackle the root causes.

[1] http://www.agilemodeling.com/essays/generalizingSpecialists.htm

Conclusions

Catalysts

- The Theory of Constraints is compatible with the Agile Movement and with Lean Thinking.
- It gives us thinking tools to identify and eliminate policy constraints.
- I have gained some really valuable insights.

"It is not necessary to change;
survival is not mandatory."

W. Edwards Deming

Questions?

Catalysts

Resources

Catalysts

- Business novels by Eliyahu Goldratt:
  - *The Goal*, 2nd Edition, North River Press, 2004.
  - *It's Not Luck*, North River Press, 1994.
  - *Critical Chain*, North River Press, 1997.
  - *Necessary But Not Sufficient*, North River Press, 2000.
- Theory books by William Dettmer:
  - *Goldratt's Theory of Constraints*, ASQ Quality Press, 1997.
  - *Breaking the Constraints to World-Class Performance*, Mc Graw-Hill Education, 1998.
- David Anderson: *Agile Management for Software Engineering*, Prentice Hall, 2003.
- Thomas Corbett: *Throughput Accounting*, North River Press, 1999.
- Lawrence Leach: *Critical Chain Project Management*, 2nd Edition, Artech House Publishers, 2005.

Appendix

Catalysts

Known Techniques from the Theory of Constraints

Catalysts

"Drum-Buffer-Rope"

- Only release work to your system if the bottleneck can process it, i.e.
  - tie the entry gate to the bottleneck with a rope;
  - introduce new requirements only at the rate that they can be implemented.
- Let the bottleneck play the drum.
- Protect the bottleneck from uncertainty with buffers; use the appropriate kind of buffer:
  - Prioritized queue (backlog) for scope uncertainty
  - Time buffer for schedule uncertainty
  - Money buffer for budget uncertainty
  - People for uncertainty with people (e.g. illness, attrition)
  - Resources for uncertainty with resources (e.g. server crash)
- Monitor buffer usage to identify trouble as soon as possible
  - Buffer usage is a leading indicator for trouble

Known Techniques from Critical Chain Project Management

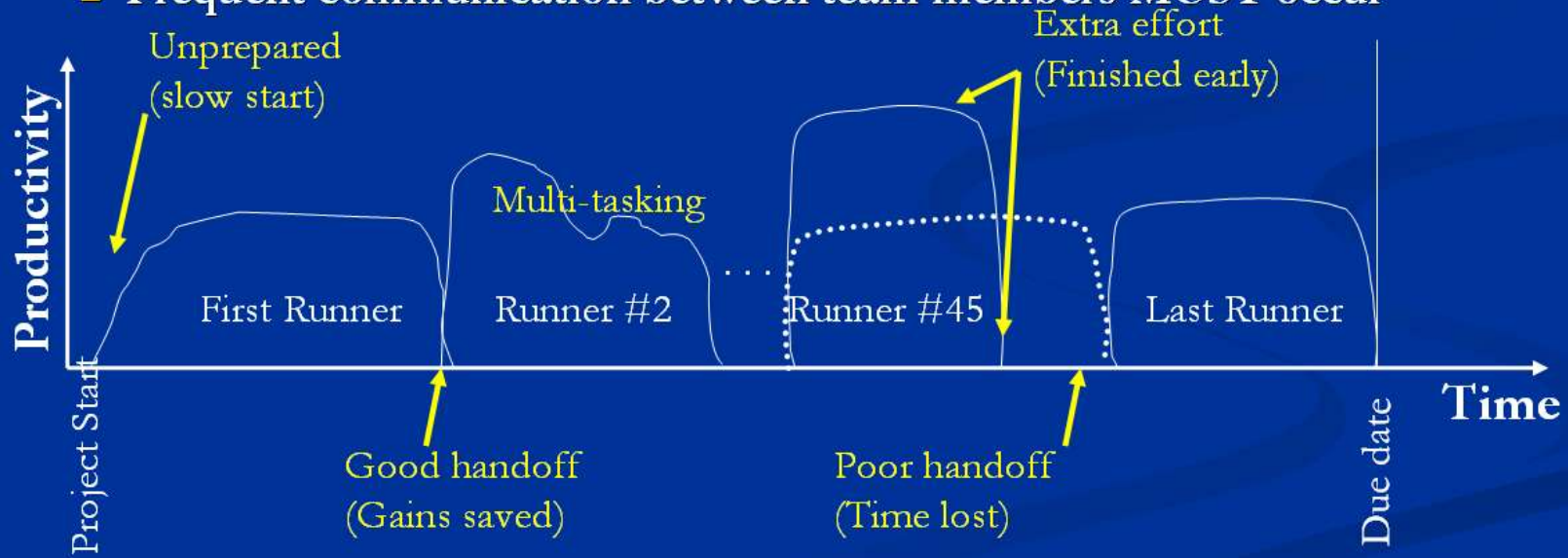
Catalysts

- **Student Syndrome**
  - Students don't start their homework early.
  - If you know that you have some spare time, you won't focus on the task at hand.
- **Multitasking**
  - If you have several things to do, every single thing takes longer to be processed.
- **Relay-Race Analogy**
  - Put the contingencies from all the task estimates into one project buffer.
  - If you hold the batton (i.e. if you are the critical resource, the bottleneck), run as fast as possible.
  - If you are not on the critical path, help the others.
  - ➔ It is okay that estimates are wrong. But it is imperative not to waste time on the critical path.
  - ➔ It's okay to be ahead or behind schedule if you inform the others, especially the next one in line.
  - ➔ Be prepared to be ready to start within 2 hours of notice.
  - ➔ Know who receives your output and regularly communicate your progress with them.

From a colleague at IBM, Douglas Buol

The CCPM Relay Race Analogy

- View each task owner is a relay runner
- When a "runner" finishes their task, a "handoff" occurs to the next
- "Delivery" occurs when the last runner crosses the finish line
- Poor preparation slows starts
- Good skills and dedicated focus accelerate completions
- Multi-tasking reduces productivity and delays completions
- Frequent communication between team members MUST occur



How do we deal with the negatives?

Focusing

Catalysts

- **Pareto Principle** (80:20 rule)
  - Solving 20% of the important problems, you'll reap 80% of the benefits.
  - The rule only applies to systems composed of independent variables.
  - The rule only applies to the cost world where each links is managed individually.
- **The rule is not applicable in the throughput world.**
  - In the throughput world, you have to focus on the ONE bottleneck (1%) and you'll reap 99% of the benefits.
  - Isn't that even better ;-?

TOC for Planning and Management

Catalysts

- Revise the Critical Path project plan with your resource constraints to get the Critical Chain project plan.
  - The critical path is based on the task dependencies.
  - The critical chain adds resource dependencies.
  - The critical chain allows to identify the bottleneck / the capacity constrained resource.
  - Don‘t waste time on critical path
  - Put all the safety at the end of the critical path in the „project buffer“
  - If non-critical paths join the critical path, add „feeding buffers“ at the end of the feeding path to protect the critical path.
  - Measure progress on the critical path only, report buffer usage

- Use Throughput Accounting for making decisions.

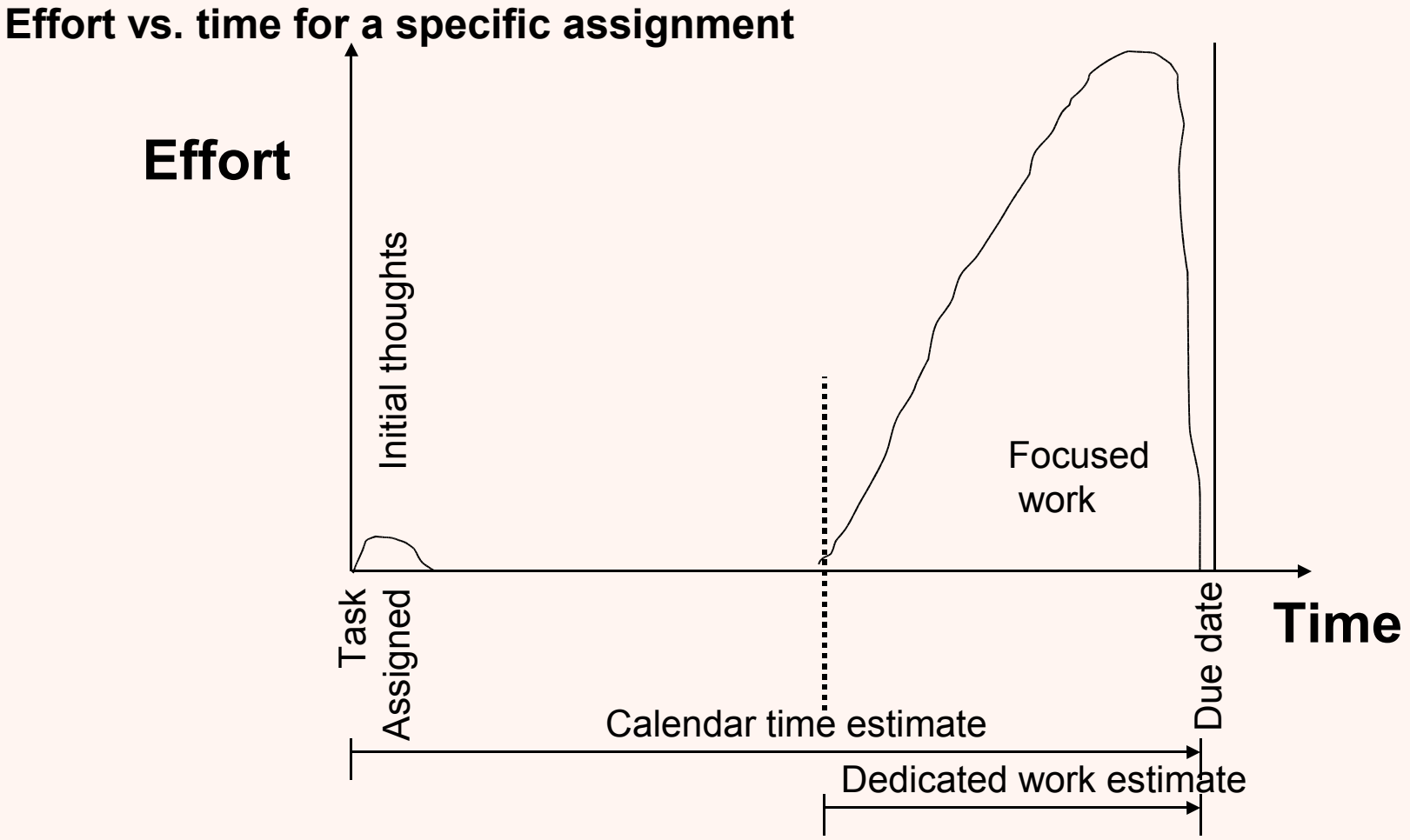
Throughput, Inventory, and Operating Expense

Catalysts

- **Throughput** is the rate at which the entire system generates money through sales. Throughput is all the money coming into the system.
- **Inventory** is all the money the system invests in things it intends to sell, or all the money tied up within the system, e.g. raw materials, unfinished goods, purchased parts, investments in equipment and facilities.
- **Operating Expense** is all the money the system spends turning Inventory into Throughput. Operating Expense is all the money going out of the system, e.g. direct labor, utilities, consumable supplies, and depreciation of assets.
- When you decide what action to take, ask yourself:
  - Will it increase Throughput? If so, how?
  - Will it decrease Inventory? If so, how?
  - Will it decrease Operating Expense? If so, how?
- If it doesn't increase Throughput, you're wasting your time, and probably your money.

The Student Syndrome

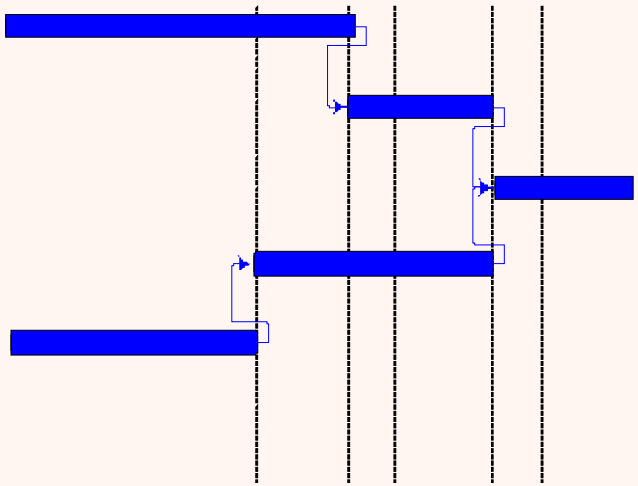
Catalysts



Local buffers are eaten up

Catalysts

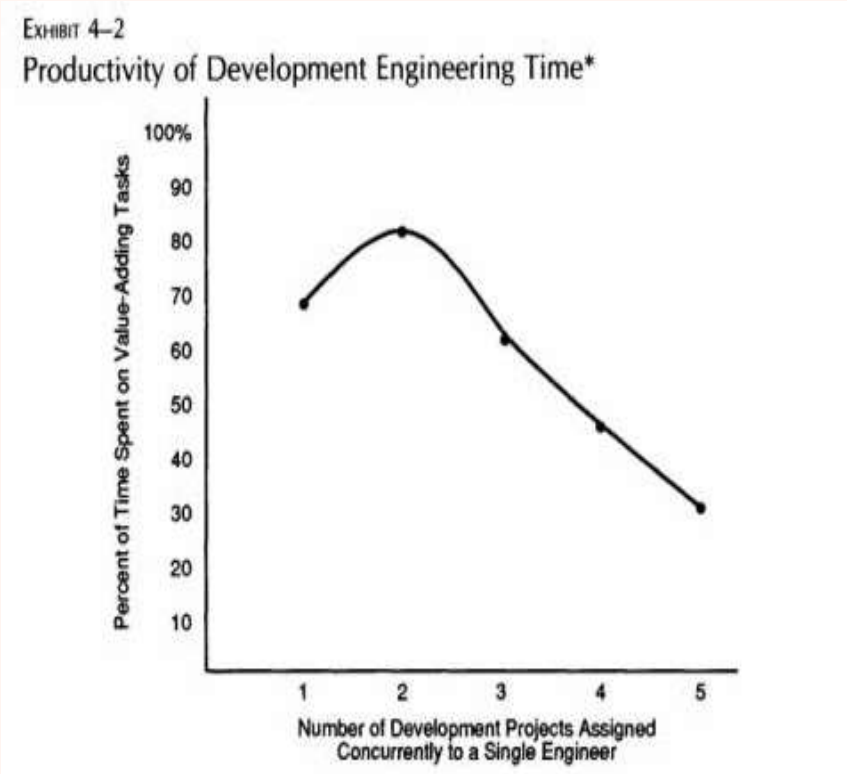
- Parkinson‘s Law: you will use up all the time you have.
- Sucessor task will be delayed as soon as one of its predecessors is late.
- Advances are not taken advantage of.



Negative effect of multitasking

Catalysts

- If you focus on one task, you are very productive, but you'll be blocked some time.
- If you can switch to another task in case of blockage, you'll increase your productivity.
- If you take on another task in parallel, your productivity decreases sharply.
- You will not have the time to focus, you will not get into the „flow".
- You will expect the next interruption and not even try to concentrate on your work.



* Steven C. Wheelwright: *Revolutionizing product development - quantum leaps in speed, efficiency, and quality*, p. 91

Eliminate Multitasking

From a presentation by Clarke Ching given at the XP Day 2004 in London

Three projects - no sharing

Resources: x3 x3 x3

P1 A B C D E F G H I J 20 weeks

P2 A B C D E F G H I J 20 weeks

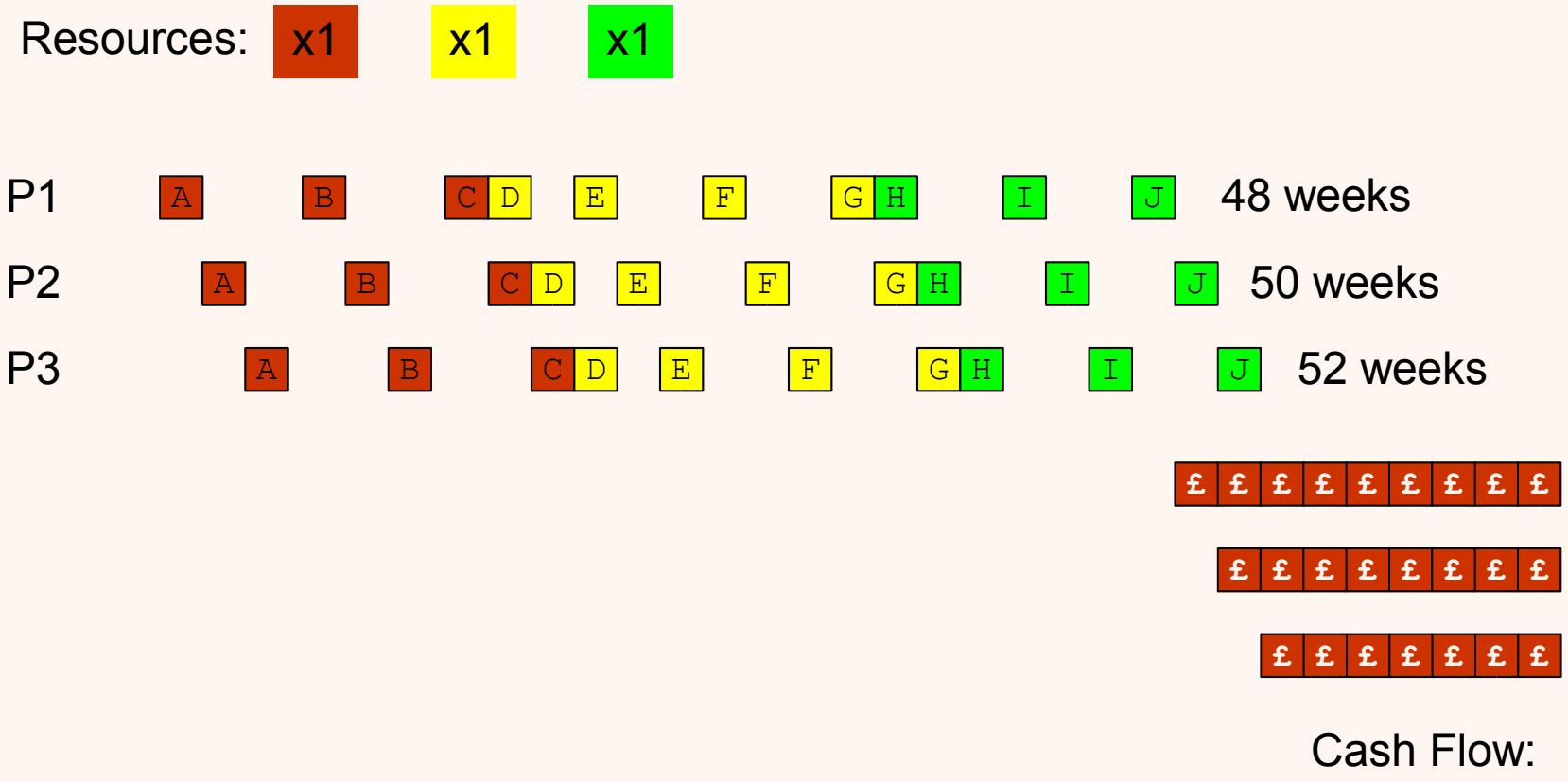
P3 A B C D E F G H I J 20 weeks

Cash Flow:

Each task takes 2 weeks to complete

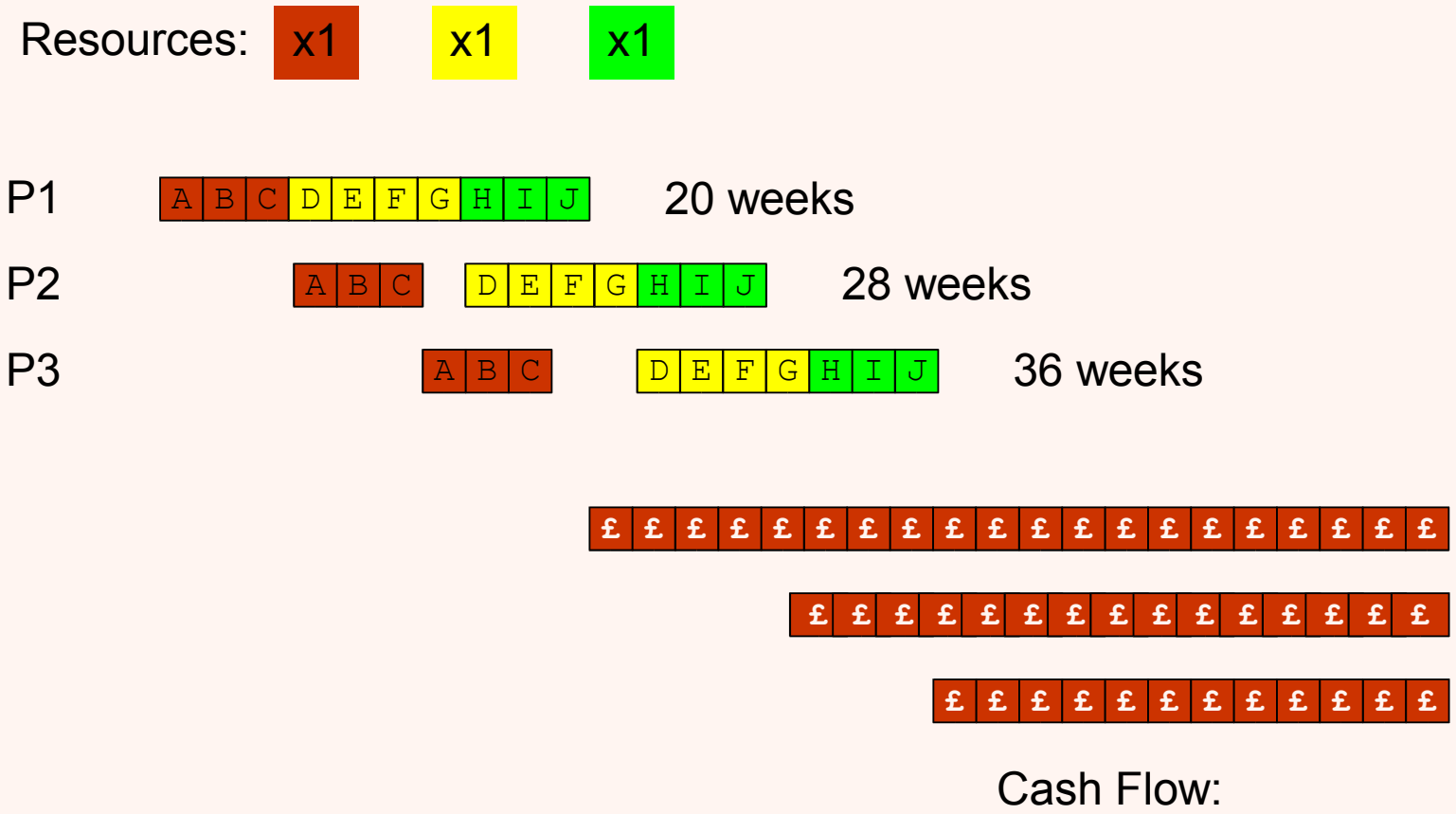
Scenario with multitasking

From a presentation by Clarke Ching given at the XP Day 2004 in London



Scenario without multitasking

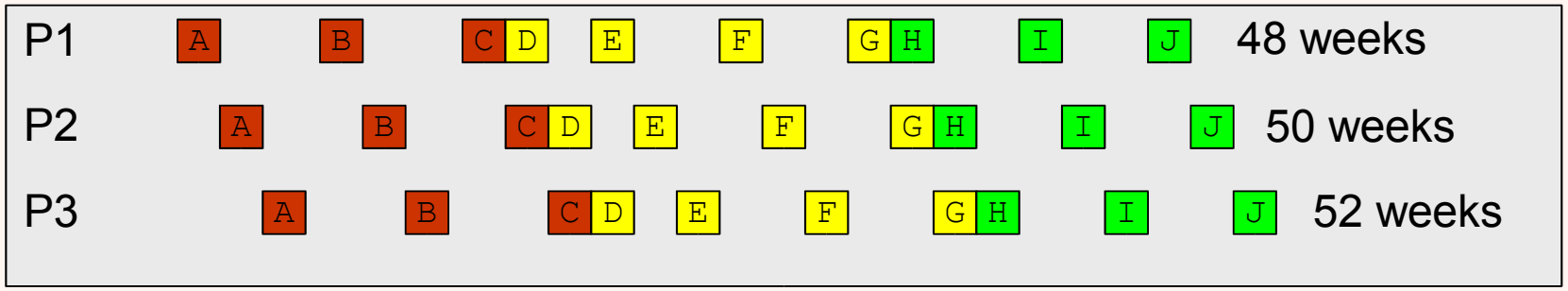
From a presentation by Clarke Ching given at the XP Day 2004 in London



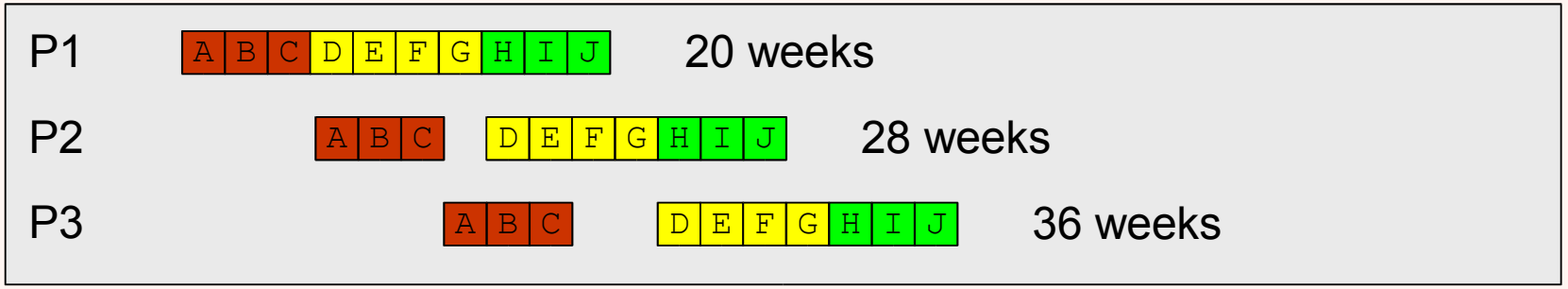
Compare

From a presentation by Clarke Ching given at the XP Day 2004 in London

With Multitasking



Without Multitasking



- Each project is shorter
- All projects finished sooner ➔capacity is increased
- Cash flow comes in earlier; faster to market, profits go up, up, up
- Organisation evolves faster

Quotes

Catalysts

- “It is not necessary to change; survival is not mandatory.” (W. Edwards Deming)
- “We must all hang together, or we shall surely hand separately.” (Benjamin Franklin during the American Revolution)
- “He who innovates will have for his enemies all those who are well off under the existing order of things, and only lukewarm supporters in those who might be better off under the new.” (Niccolò Macchiavelli)
- “Courage is the power to let go of the familiar.”
- “Learning usually passes through three stages. In the beginning, you learn the right answers. In the second stage, you learn the right questions. In the third and final stage, you learn which questions are worth asking.”
- “Tell me how you will measure me and will I tell you how I will behave.” (Eliyahu Goldratt)
- “The detail is meaningless until you understand the paradigm.” (Michael Kennedy)