



THE ITR GROUP_{LLC}
Information Technology Resources

Repeat IT Project Success

Using a Standard SDLC Process
- System Development Life Cycle -

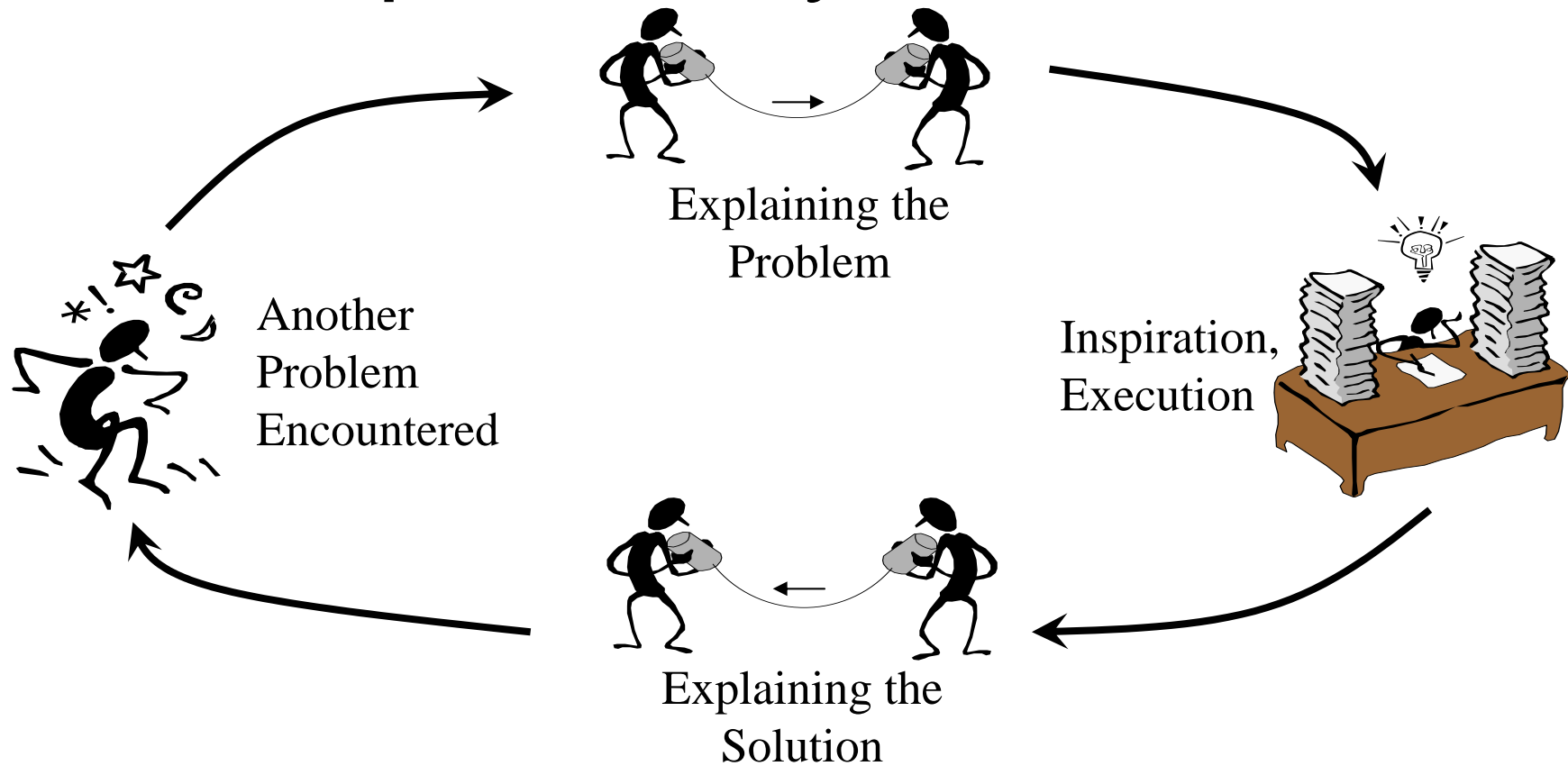


The Theme

Using a standard system development lifecycle, you can:

- Repeat previous successes
- Save money and time
- Improve system quality
- Enhance your ability to deliver
- Ensure expectations are met

Projects without a standard: The repeatability of failure



The endless circle: Start anywhere, and keep going!

Projects without a standard: Common complaints

That's NOT
what I asked
for!

That's what I
asked for, but
it's not what I
want!

Why does it
take so long to
run?

Sorry to wake
you, but it blew
up again. Now
what?

Can you
remember how to
change the report
format?

Projects without a standard: The Aftermath



- Production outages & performance problems
- “Close-enough” solutions that miss the mark
- Hard-to-maintain code & processes
- Word-of-mouth system documentation
- Heroic efforts required to keep systems running
- Critical resources bogged down in support

Result: Inefficiencies and Opportunity Costs!

Projects without a standard: Aren't they cheaper?



- Lack of Quality means Loss of Productivity
 - ✓ “Close enough” means expensive rework
 - ✓ Rework means no resources for new projects
- Concentrated expertise in undocumented systems means increased Business Risks
 - ✓ Vulnerable to staff attrition
 - ✓ Held up by staff demands
 - ✓ Scheduling around staff bottlenecks

As responsible managers, we must not let this happen!

Projects without a standard: Are they unusual? (apparently not)



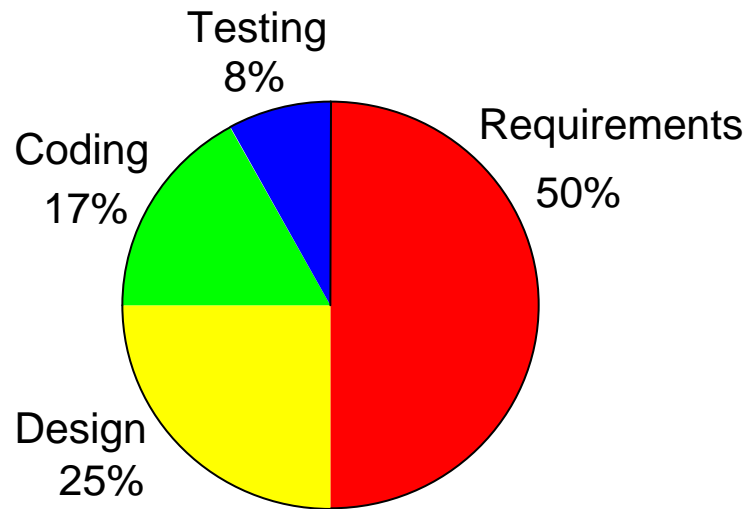
- Some 40% of systems projects fail before completion
- About 50% of systems projects overrun their budget by 180% or more
- Cost of failed systems projects amounts to more than \$100 billion annually
- The list goes on ...

“Not unusual” is not the same as “acceptable”!

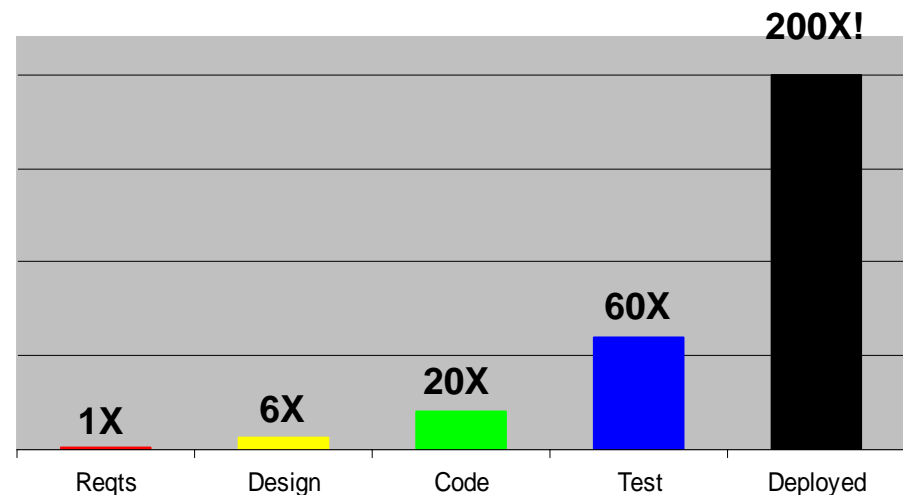
Where do the problems arise?

And when is it best to find and correct them?

Sources of Errors

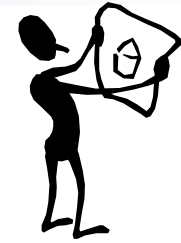


Relative Cost to Correct Errors



The earlier we find and correct errors, the better!

How do we get there?



By using a **System Development Life Cycle (SDLC)**, with **standard templates, standard processes, and standard framework to structure how you can:**

Frame the problem

Project Charter, Scope Statement

Figure out what to do

Functional & Technical Specifications

Figure out how to do it

Architecture & Design

Do it

Implementation: Code & Unit Test

Verify it was done correctly

Testing & Formal Signoff

Deploy it to production

Deployment

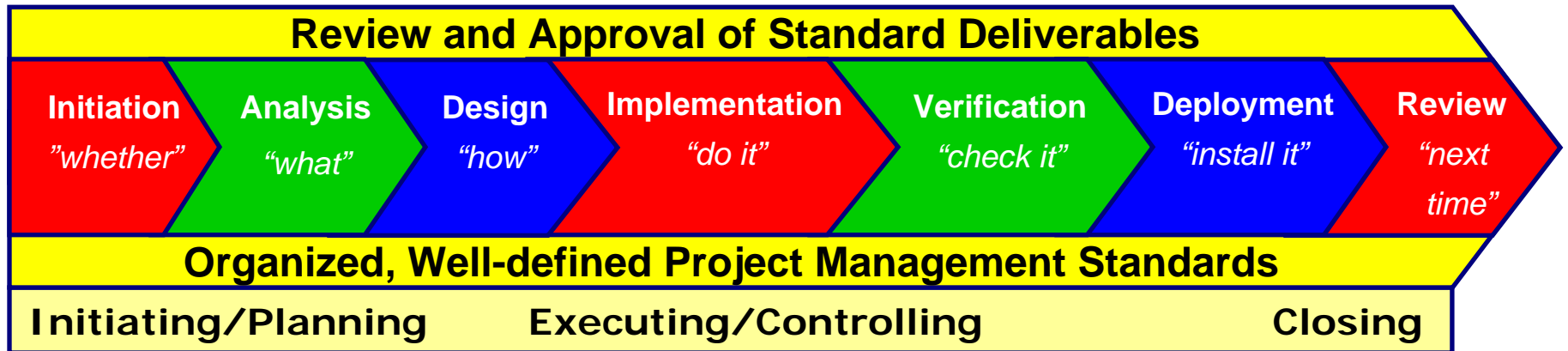
Improve the process

Post-Deployment Review

Use frameworks such as PMBOK & SEI's Capability Maturity Model Integration (CMMI) to build your own, or find a pre-defined SDLC such as Rational Unified Process (RUP) or Agile Product Lifecycle Management (PLM) – but get one somehow and use it!!

*If the problems sounded familiar, **this** is your way out!*

What are the components?



Initiation

Is this the right thing to do?

Analysis

What exactly is needed?

Design

How can we make this work?

Implementation

Getting the job done!

Verification

Making sure it is done right!

Deployment

Getting it into use!

Review

How can we do it better next time?

Each project has a beginning, a middle, and (especially) an end!



Initiation Phase:

Is this the right thing to do?

- Establish business case
- Define project scope
- Estimate project schedule, resources, and budget
- Identify project constraints & assumptions
- Define project organization and controls
- Prepare statement of work

PMBOK: Initiating Process Group



Business Requirements:

What exactly is needed?

- Identify opportunities for process improvement
- Evaluate alternative approaches
- Define the functional requirements
- Consider iterative delivery possibilities

PMBOK: Planning, Executing, and Controlling Process Groups



System Design:

How can we make this work?

- Iterate design & prototyping
- Build and exercise application prototypes
- Define technical architecture
- Develop application architecture
- Create system and integration test plans

PMBOK: Executing and Controlling Process Groups



Implementation:

Getting the job done

- Develop detailed system specifications
- Create the actual system code

Use iterative build/test approach:

Execute construction, system testing and integration testing in multiple parallel cycles

- Unit test the code as developed
- Test all changes together when done

PMBOK: Executing and Controlling Process Groups



System & User Acceptance Test:

Making sure it is done right

- *System Test*: Formal structured test of the completed system or system changes
 - ✓ Functional tests: both positive and negative
 - ✓ Performance/load/stress testing
 - ✓ Scripted, repeatable, and re-usable
- *User Acceptance Test*: Official “test drive” of system, procedures, and documentation by business users - before deployment!

PMBOK: Executing and Controlling Process Groups



Production Deployment:

Getting it into use

- Establish the production environment
- Install application components
- Convert data if necessary
- Train users and operations staff as needed
- Switch operation to the new application
- Move to Support mode

PMBOK: Executing and Controlling Process Groups



Post-Implementation Review:

How can we do it better next time?

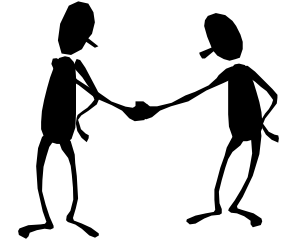
- Process Improvement
 - ✓ What went right - how can we repeat it next time?
 - ✓ What went wrong - how can we prevent it next time?
- Product Improvement
 - ✓ How was the end result received?
 - ✓ What are the follow-ups, if any?
 - ✓ Do we need to initiate a new project?

PMBOK: Closing Process Group

How do we make this happen?

The “CONTRACT”:

- **Commitment** by all participants
- **Organization** of the project and the team
- **Negotiated agreement** on scope and schedule
- **Teamwork** among all participants
- **Routine** application of these principles
- **Accountability** on all sides
- **Coordination** among all involved parties
- **Timely** completion of promised deliverables



Where do I sign?

References



THE ITR GROUP_{LLC}
Information Technology Resources

- *Project Management Institute: (PMBOK)*
Project Management Body of Knowledge
www.pmi.org
- *Software Engineering Institute: (SEI/CMMI)*
Capability and Maturity Model Integration
<http://www.sei.cmu.edu/cmmi/>
- *Most importantly, your best resources are internal:*
Your own people understand how they do their jobs. Structure created from existing processes is much more readily accepted than structure imposed from above!