Developing IT Business Solutions in a Regulated Environment

Specific Challenges of Smaller Pharmaceutical and Biotech Companies

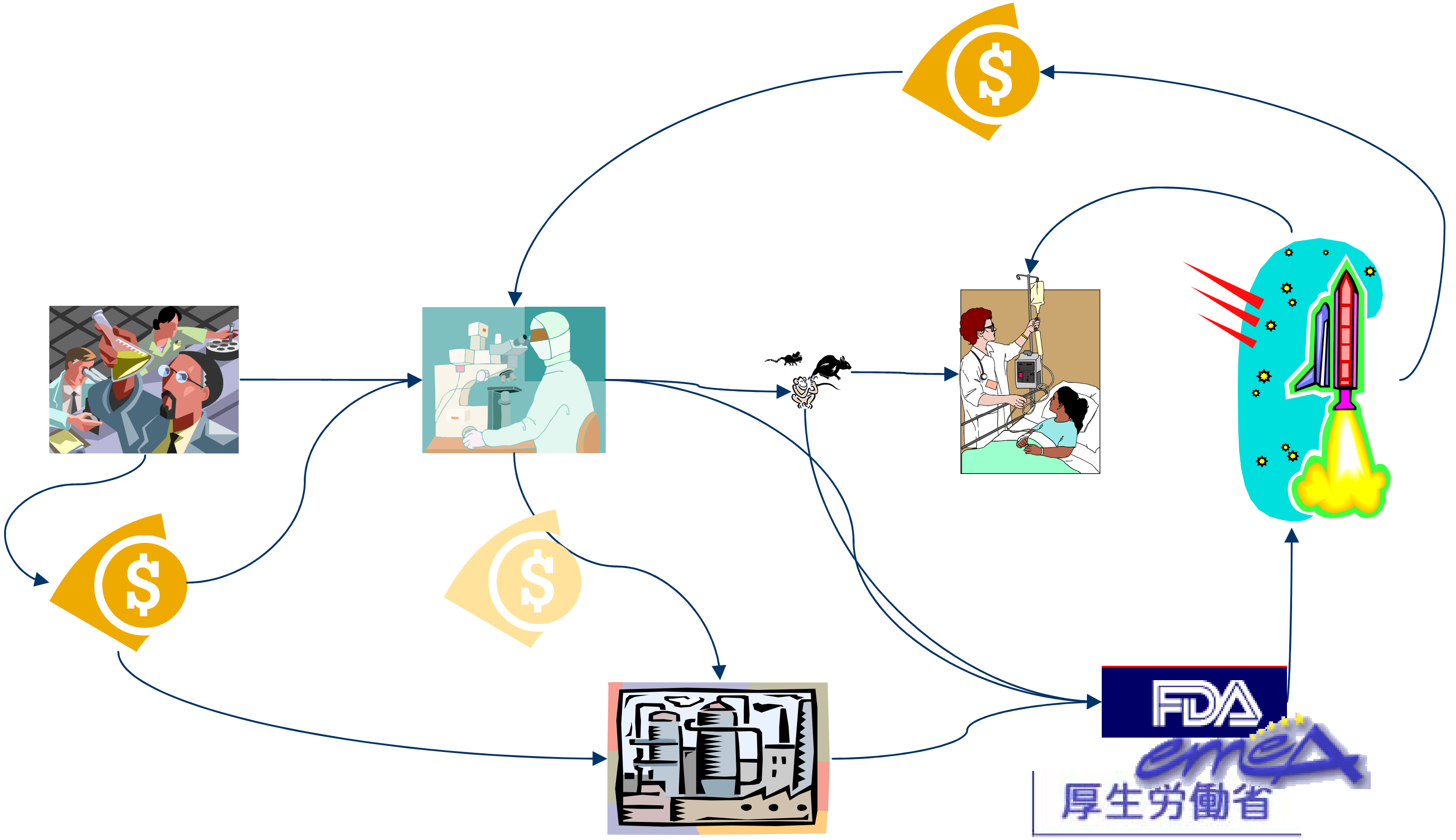
PRO
sys

What we want to talk about

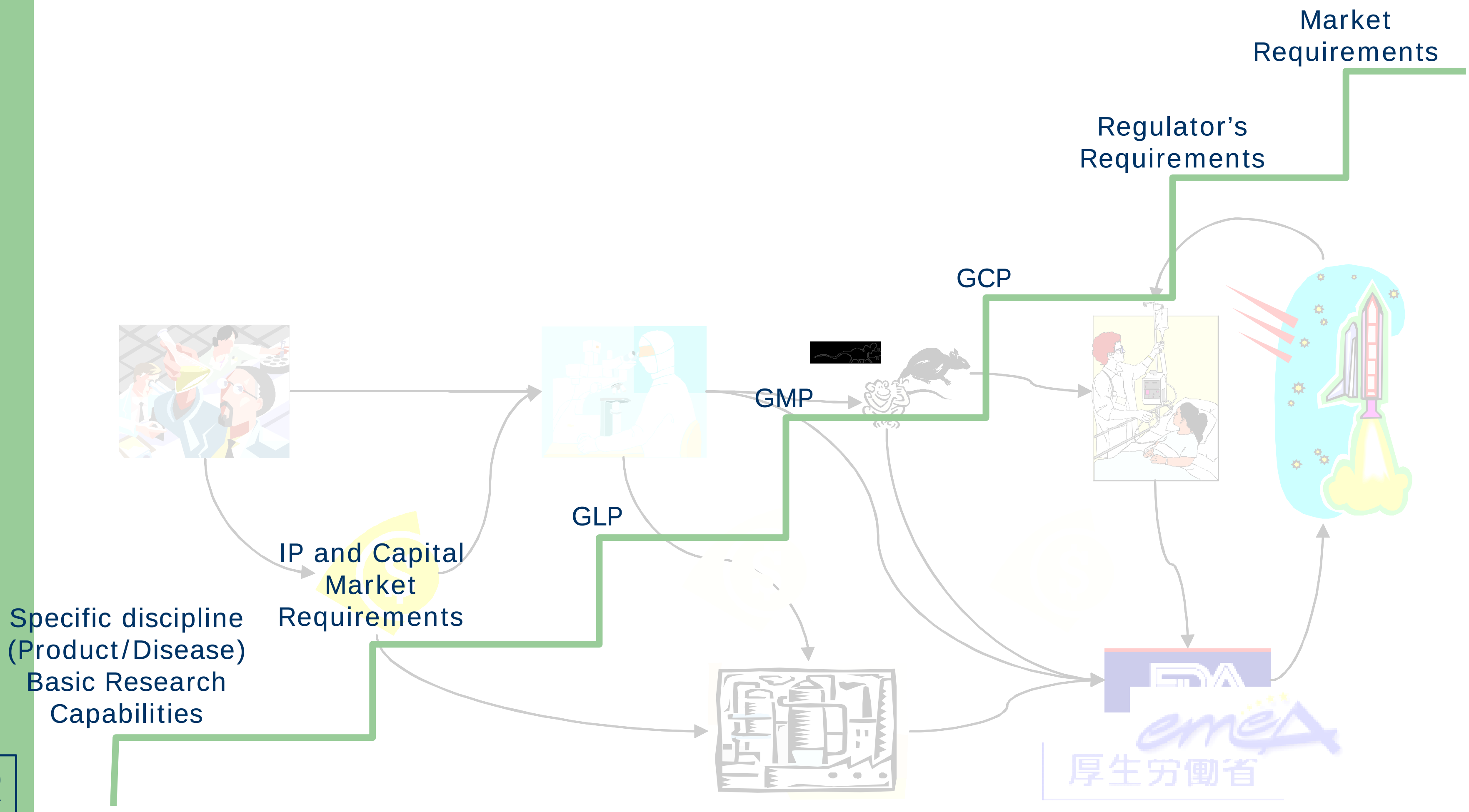
- On being/becoming a Biotech/Pharmaco
- What role data and document management systems play in business solutions
- What (not) to do when deploying business solutions

PRO
sys

Being/becoming a Biotech/Pharma…



…is about having/building capabilities…



Specific discipline (Product/Disease) Basic Research Capabilities

IP and Capital Market Requirements

GLP

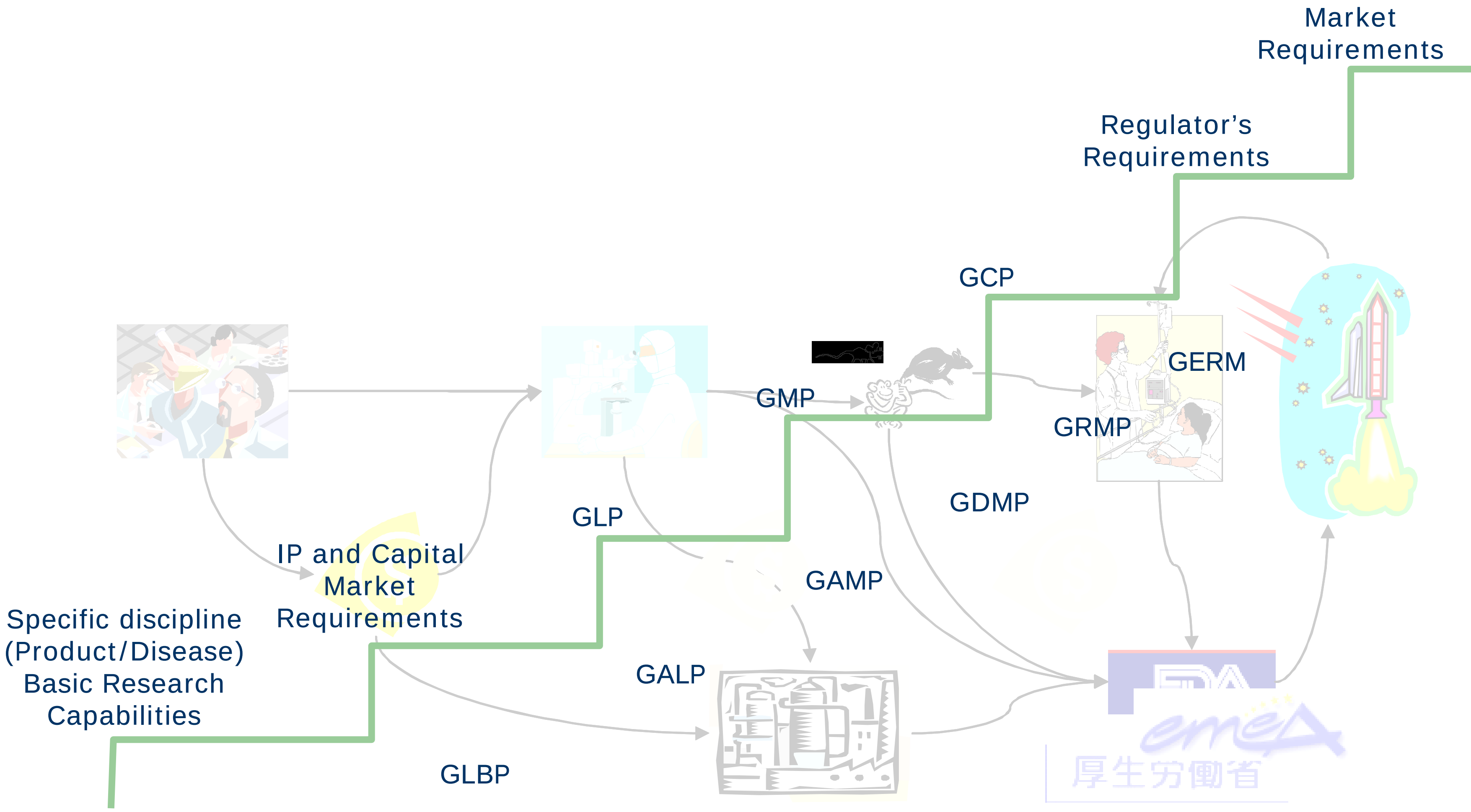
GMP

GCP

Regulator’s Requirements

Market Requirements

...working to good practices



Specific discipline (Product/Disease) Basic Research Capabilities

IP and Capital Market Requirements

GLBP

GLP

GALP

GAMP

GMP

GCP

GDMP

GRMP

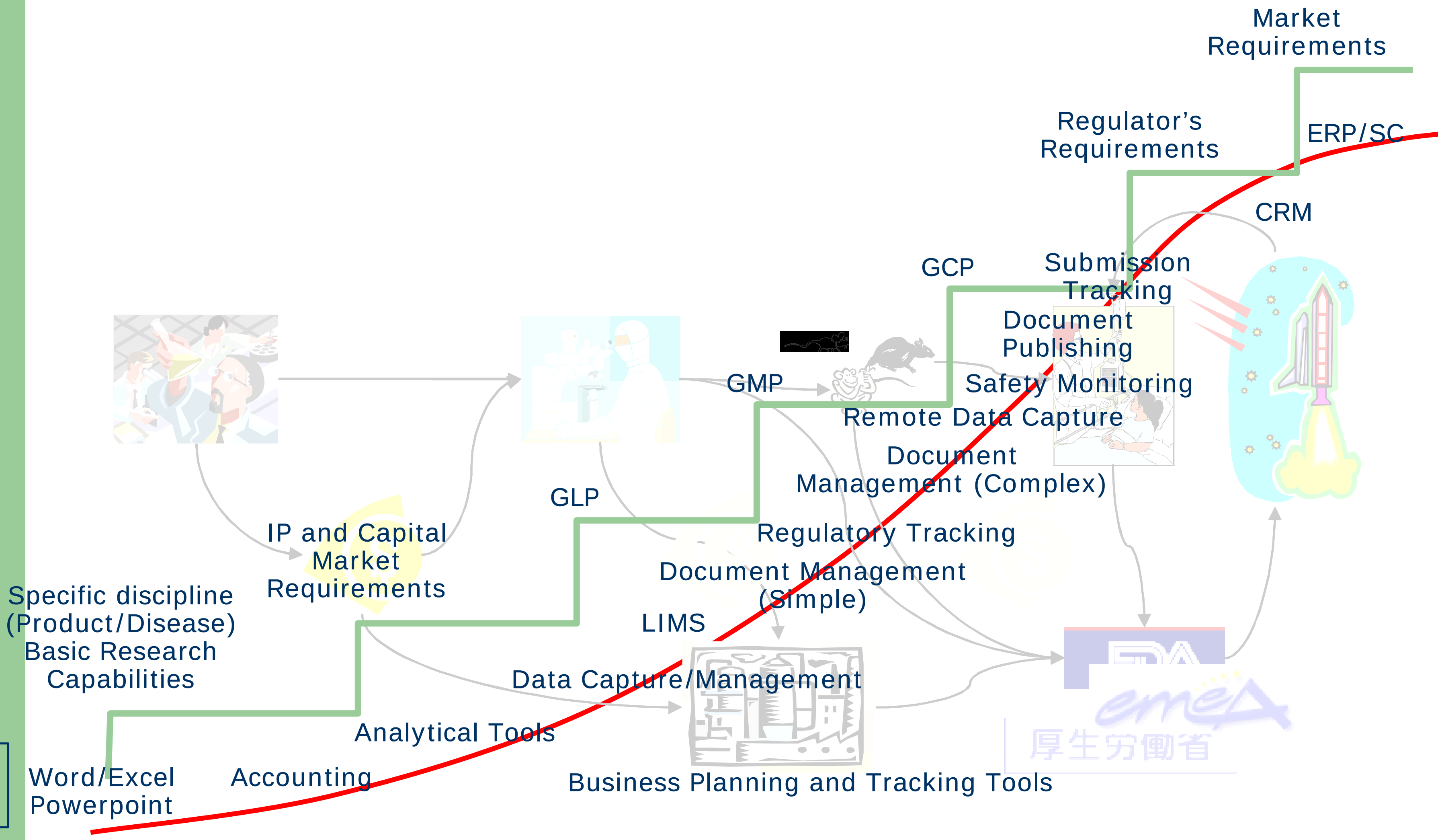
GERM

Regulator's Requirements

Market Requirements

PRO sys

Information systems help us get there



Market Requirements

Regulator's Requirements

ERP/SC

CRM

GCP

Submission Tracking

Document Publishing

Safety Monitoring

Remote Data Capture

Document Management (Complex)

GMP

GLP

Regulatory Tracking

Document Management (Simple)

LIMS

IP and Capital Market Requirements

Specific discipline (Product/Disease) Basic Research Capabilities

Data Capture/Management

Analytical Tools

Word/Excel Powerpoint

Accounting

Business Planning and Tracking Tools

LSO* Core vs. Non-core business

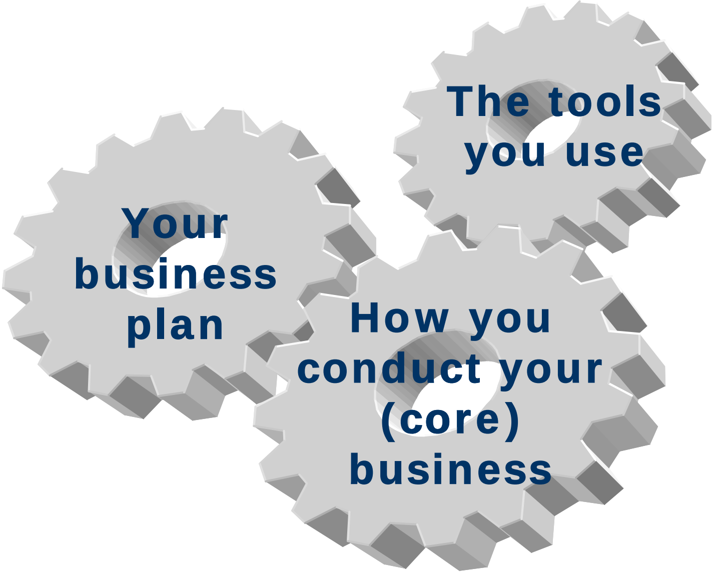
- The core business of LSOs is Life Sciences:
  - Identifying and studying molecules, genes, and proteins
  - Analyzing and interpreting the data generated
- Your business plan determines whether manufacturing, clinical development, sales, are part of your core business
- Defining/Selecting/Building data and document management systems is <u>NOT</u> the core business of LSOs

PRO
sys

* Life Sciences Organization

Where systems fit

- The ability to define how you conduct your core business should be a core competency
- Defining and identifying the best way to conduct your core business should be a core competency

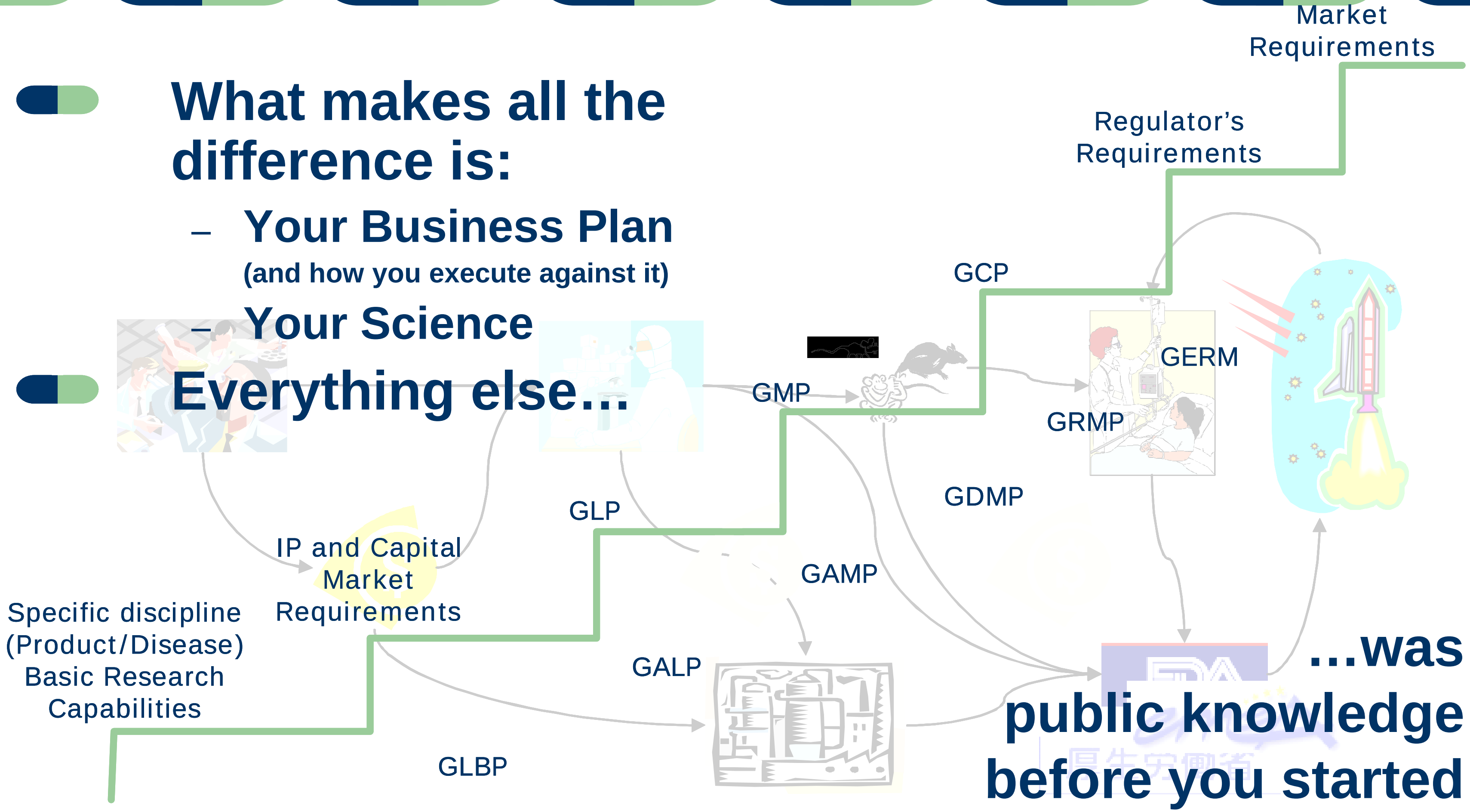


- Information systems support (or hinder) the way you want to conduct your (core) business

So what should be proprietary

- What makes all the difference is:
  - Your Business Plan
    (and how you execute against it)
  - Your Science
- Everything else…

…was public knowledge before you started



Market Requirements

Regulator's Requirements

GCP

GERM

GMP

GRMP

GDMP

GLP

GAMP

IP and Capital Market Requirements

Specific discipline (Product/Disease) Basic Research Capabilities

GALP

GLBP

Where does the problem begin?

- Small LSOs tend to fall into the first trap, and leave the deployment of supporting tools until too late
- Large LSOs tend to fall into the second trap, and become software houses

PRO
sys

Where does the project begin?

- Day 1: New LSO:
  - Financial systems – easy:
    - No doubt you tested your system. Or you put a few procedures in place
    - Remember, most financial systems don't go beyond simple arithmetic
  - Other system – defer until necessary
- Day n: new development phase:
  - Too late to put the system in place for that phase
  - You should have had a plan
  - Revisit Day 1 – planned deferrals are ok

The project is part of a Master Plan

- “A plan never survives first contact with the enemy”:
  - Don’t hold the plan sacred – let it serve you
- Modern planning tools allow and encourage the document to live and change over time
- The Master Plan is an evolving document which helps you govern your next steps:
  - It ties in with high-level business events and objectives

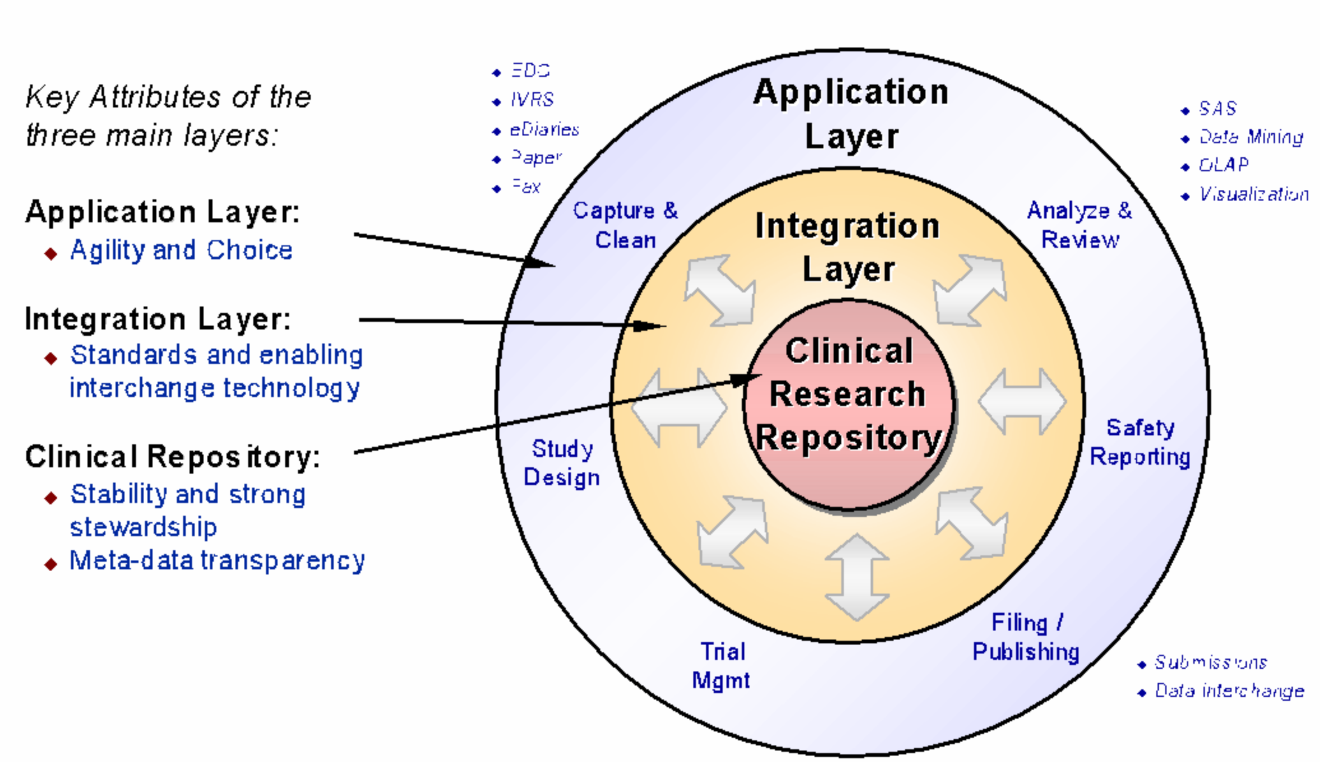
PRO
sys

The Master Plan is a project

- Don’t underestimate the value:
  - Don’t underestimate the effort either
- The (whole) company needs to own the plan:
  - Understand it:
    - It needs a communication strategy that maintains awareness of the plan throughout its lifetime
  - Believe it
  - Live it
  - Sign-off on it:
    - More than just ink
    - It is a commitment
  - Monitor progress against it

Do you have an Architecture?

- A conceptual framework of standards that all components have to fit into



Key Attributes of the three main layers:

Application Layer:
- Agility and Choice

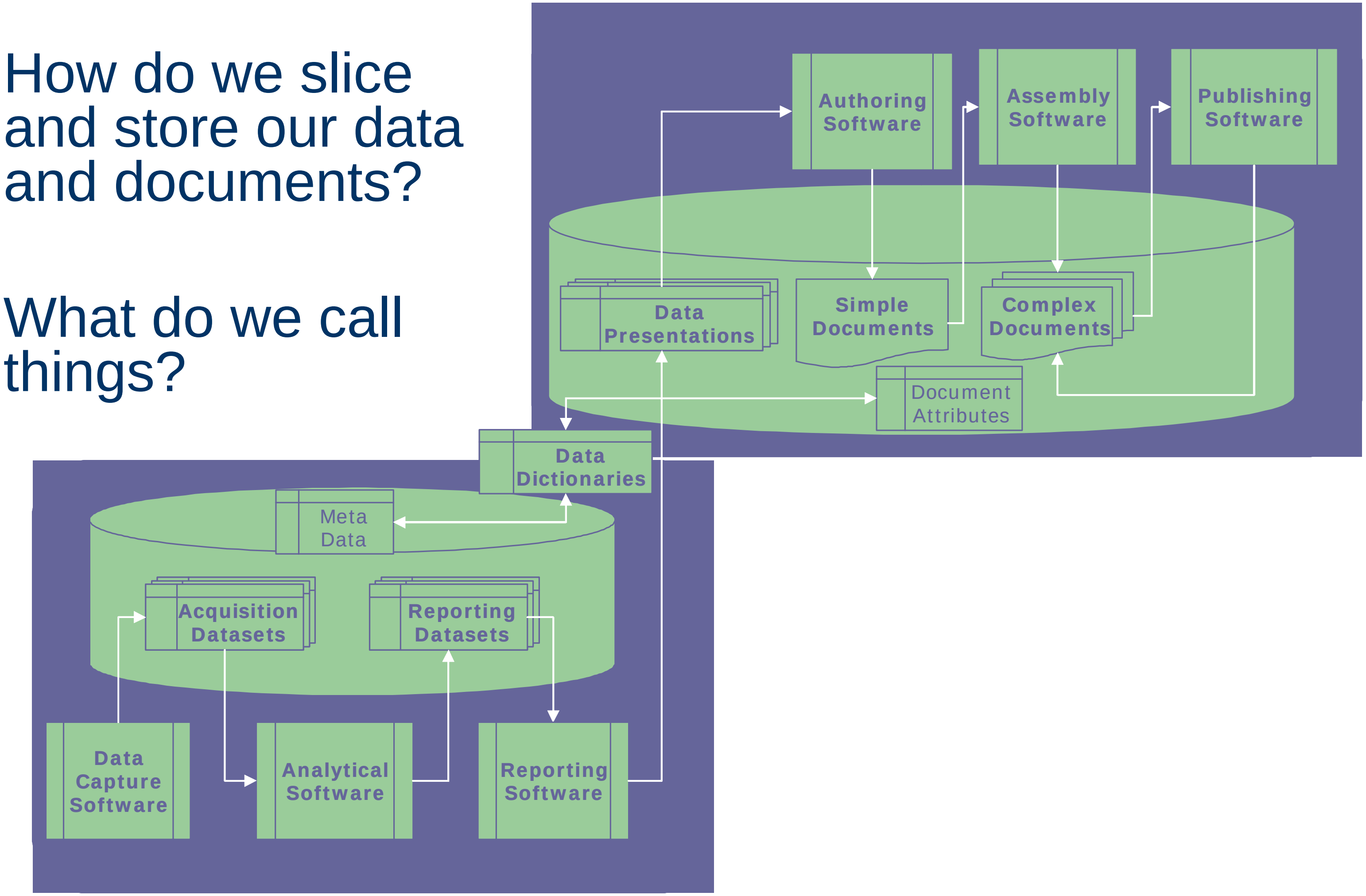
Integration Layer:
- Standards and enabling interchange technology

Clinical Repository:
- Stability and strong stewardship
- Meta-data transparency

Architectures support integration!

- How do we slice and store our data and documents?

- What do we call things?



PRO sys

Requirements Definition

A project has a

[beginning

... a middle ...

and an end]

- But how will you recognize these things?
- Do you need “perfection” or “good enough”?
- Many projects fail for not knowing when they’re done!

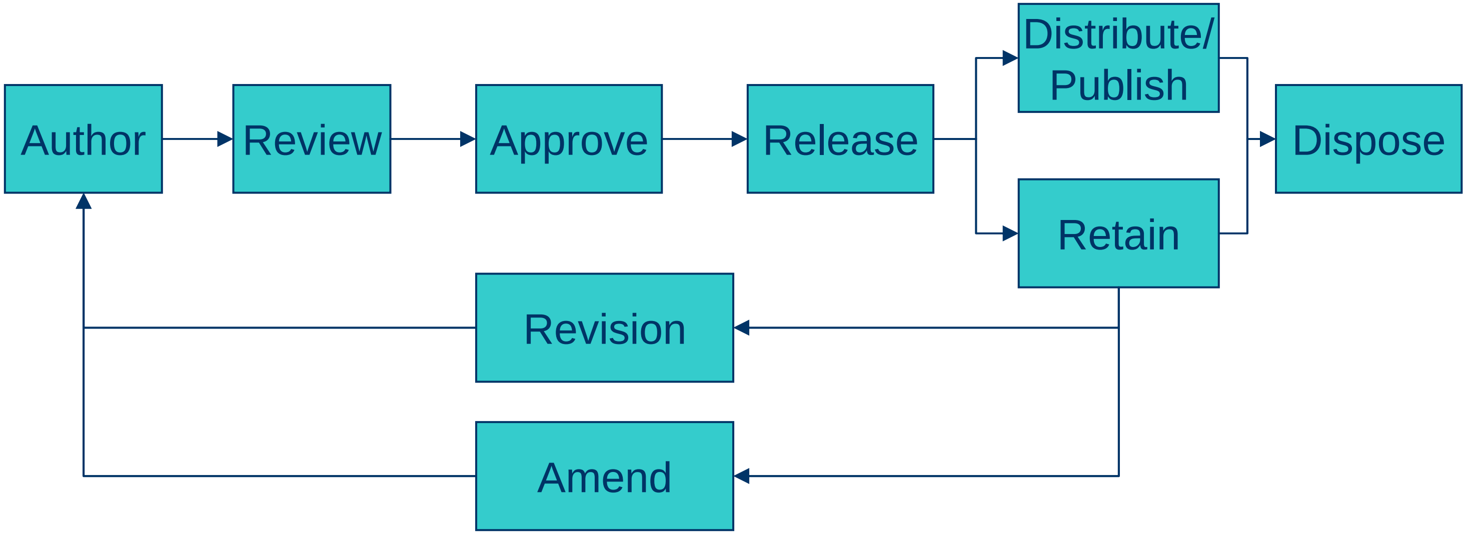
PRO
sys

Requirements Definition

- Define the end at the beginning:
  - Distinguish between requirements for the solution and requirements for the system
- Until you’ve defined the end, the beginning is not finished
- Justify the end:
  - “The system will have a blue flashing cursor” vs. “The system will have a blue flashing cursor (see 21 CFR Part F.22.1)”
- Define “good” requirements:
  - Complete, <u>testable</u>, unambiguous, etc.

Focus on the Generic Pieces

- There are many types of documents
- There are only a few types of processes that address the needs of all document types



- Deploy systems capabilities that support the generic processes to get the best return on your investment

PRO
sys

Who are actually the stakeholders?

- A poll of participants usually shows that the “business users” are considered to be the main stakeholders:
  - Hence requirements are commonly called “user requirements”
- In reality, there is a higher end to be served:
  - Corporate requirements need to be catered to as well
  - You may not validate against them, but should certainly test and measure the result against them
- Example:
  - A document management system has very few redeeming features for end users when compared with alternatives such as corporate email
  - The corporate requirement may nonetheless be that DMS use is mandatory

Don’t seek the solution where the problem is

- Often the solution to the problem is found upstream and downstream of the problem:
  - We don’t know how to index our documents:
    - What do you need to know about them?
    - What do your clients need to know about them?
    - What do the originators know about them?
  - How do we manage completeness of our files?
    - We know what we have
    - We often don’t know what we don’t have:
      - Bring the originators into the solution:
        - Remember, the whole company owns the plan

Development vs. Selection

- a.k.a. “Build vs. Buy”
- You don’t build your own desks and workstations:
  - But you probably bought them early on
- Even the US military doesn’t build their own stuff:
  - They buy it when they anticipate the need
- Challenge:
  - Does anyone here have a proprietary business model for pharma / biotech?
    - If you have one, I’ll sign a non-disclosure and buy you dinner to find out what it is!

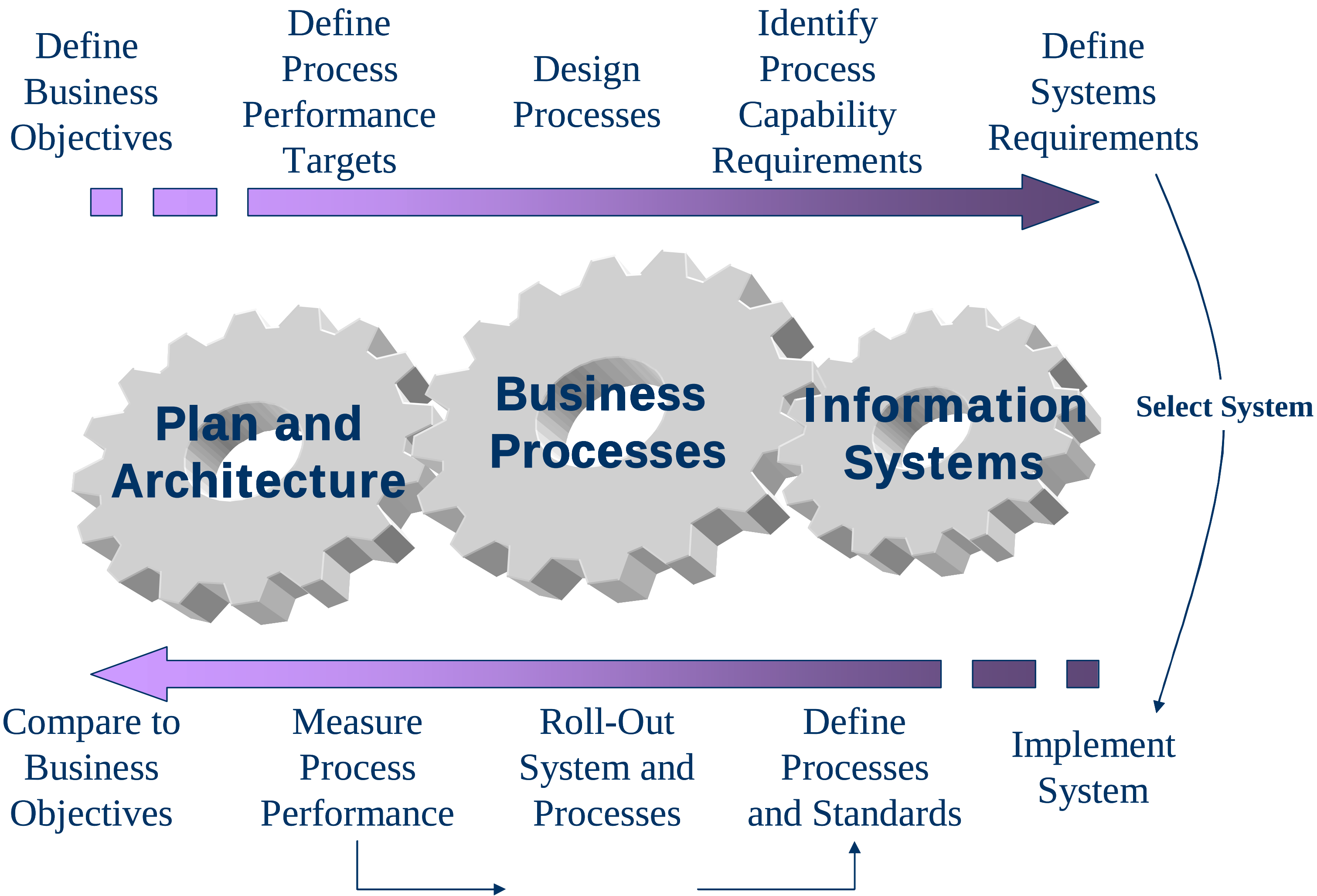
PRO
sys

The customization trap

- Most LSOs believe they have specific unique requirements
- Most consultants would like these LSOs to continue to believe it
- First rule: you aren’t as unique as you think. Don’t customize
- Second rule: even if you are unique, you are only trying to address 90% of your needs with a business system. The 10% probably includes your unique bits. Still don’t customize
- Look for configurability – Don’t customize

PRO
sys

Two stages to a business solution



Define Business Objectives

Define Process Performance Targets

Design Processes

Identify Process Capability Requirements

Define Systems Requirements

Plan and Architecture

Business Processes

Information Systems

Select System

Compare to Business Objectives

Measure Process Performance

Roll-Out System and Processes

Define Processes and Standards

Implement System

PRO sys

Detailing the plan

- There is no need to invent it from scratch
- The minimum steps are clear and defined by the regulations
- The minimum deliverables are defined, and templates are available
- There is a long list of public and inexpensive sources of individual bits and pieces
- The number of viable vendors for key systems is actually quite limited:
  - If you aim at the generic pieces
- Tailoring it to your current situation and business strategy is where you ensure that the plan delivers only what your specific strategy requires:
  - Why buy and validate SAS if your strategy is to outsource analysis and reporting of clinical data. That was an expensive toy!

PRO sys

Working the plan (1)

- Now you've defined your end, you can start
- Work the plan, but re-visit the plan:
  - Keep an eye on the objectives. If they change, you have to raise a flag
- Do not hold the plan sacred, but stick to it between revisions
- Maintain the plan so that it:
  - Shows in detail, what you can realistically foresee:
    - Six to twelve weeks?
  - Shows in general, what you would like to happen after that:
    - Six to twelve months?
  - Helps you meet your targets:
    - If not, the plan or the targets are wrong. Revise.

Working the plan (2)

- For individual deliverables, measure what is delivered against the requirements:
  - A glossy brochure may “need” to be grammatically perfect
  - A systems-design document probably does not
- Requirements, by the way, speak to purpose:
  - What’s this deliverable for?
  - Maybe its requirements are good enough, then..
- Documents are deliverables, not just flashy screens:
  - Manage expectations

Implementation

- We never said don’t integrate. We said don’t customize. You need to integrate
- Don’t let integrators customize in the guise of integration
- Right-sized integration: getting something standard to talk to something else standard:
  - You’re unlikely to be the first. But you never know

Technology

- Usually, technology components should only be adopted if they help meet the requirements as understood
- Don’t spend too much time on technology, platforms and infrastructure questions:
  - It just distracts from the original problem
  - Decide, then live with it
- We all know that technology shouldn’t lead the business, however…
  - In an age of rapid technological change, it is more acceptable than before to evaluate technology to see if it changes your understanding of the business requirements:
    - But measure the benefit in terms of business impact

Standards

- Standards are worthwhile:
  - If widely accepted, technology components and resources will be available which work with them
  - They enable and simplify integration
  - They eliminate customization
  - They encourage vendor competition
- Government is more able to publicly endorse standards than proprietary means:
  - Compliance is easier
- THE Standard doesn’t exist
- You still need to define some of your own:
  - Remember, others have done it before
  - Chances are, they weren’t all wrong
- Again, don’t spend too much time on it:
  - Any standard is better than no standard
  - They can always be improved incrementally

Processes

Defined/Documented Processes

+ Defined Standards

+ Trained Staff

+ Tested Applications

= A Business Solution (a.ka. A System)

- A process is a set of steps that adds value in a defined way: it needs requirements of its own:
  - If a process adds no value, it doesn't matter how many steps it has… get rid of it!
- Attack the right process:
  - Don't build a process because another one doesn't work. Fix the other one!

Organization

- The Organization should support the process:
  - Don’t fragment the process to enable the organization
- Keep it simple:
  - Someone needs to set Standards
  - Someone needs to execute to standards
  - Someone needs to test and document whether standards have been followed
- It doesn’t have to be real
  - Virtual often scales faster and cheaper
  - You can always make it real (bring it in) later, when you have learned
- Remember the cost of your talented resources:
  - The opportunity cost by far outweighs the $-cost

Execution in a regulated environment

- In any normative business environment, you test your systems against your needs (requirements)
- The truth is that LSOs usually grow up for a long time not being normative, but businesses based on pure science:
  - Scientific method was enough

Execution in a regulated environment

- LSOs who start late have to revisit every one of their existing systems, and create a remediation plan which they call a "Master Validation Plan":
  - It is usually at variance in terms of resources and schedule with previously planned business events
  - Consequently, compliance is seen as a barrier to business, rather than a slipstream to market

Execution in a regulated environment

- Testing against requirements (a.k.a. “validation”) is not a complex or necessarily onerous task:
  - Say what you’re going to do:
    - Document your requirements and write a plan to say how you’ll verify they are achieved
  - Do it:
    - Deploy and test your system, recording your results as you go, according to your plans/protocols
  - Demonstrate that you did what you said:
    - Document your conclusion that your results demonstrate achievement of your requirements

Execution – escape hatches

- Do not be afraid to commit when writing your plans: failure to achieve a requirement is not necessarily complete failure:
  - But record why you:
    - A) dropped the requirement; or
    - B) altered the requirement; or
    - C) achieved the requirement in an unplanned way; etc.
- The penalty for non commitment is harsher than for committing to the "wrong" thing:
  - The intent is honored
- Some requirements are non-negotiable:
    - How they are achieved however, is often very negotiable

Special considerations for “young” LSOs

- Don’t fall into the “leave it ‘til later” trap:
  - Commission a Master Validation Plan based around standard LSO lifetime events:
    - It’s a living document, just like a personal financial plan, revisit it periodically
  - The sooner you start, the more aligned it will be to your Master Business Plan:
    - In fact, it can be the Master Business Plan

Outsourcing vs. in-house responsibilities

- The usual Occam's Razor applied to outsourcing vs. in-house responsibilities and relates to non-core vs. core activities
- For an LSO, information management is **the** core activity:
  - Deploying (but not building) business solutions is therefore also core
  - Understanding deployment processes and matching user and corporate requirements is therefore also core

Outsourcing vs. in-house responsibilities

- However, for these core requirements, the skill-sets are scarce commodities
- For non-core requirements, the skill-sets are less scarce:
  - Consequently, LSOs tend to focus on the skills they can easily hire in-house, leading to the “build” trap of larger LSOs
- For small LSOs, even some core competencies can usefully be outsourced, if the process involved is “mainstreamed”

PRO
sys

Outsourcing vs. in-house responsibilities

- Recommended model:
  - Have someone internal who own the MBP/MVP.
  - Engage external resources to help create, maintain and audit adherence to the MVP (at least)
  - Such external resources should be able to:
    - Articulate a plan that maps to LSO lifecycle events
    - Help select and deploy business information systems at the appropriate time, by way of program management, project management and procurement / deployment skills
    - Have a wide-ranging industry-focused knowledge
    - Understand the regulated aspects of the LSO space

Are these projects ever complete?

- Yes, as individual projects, if…
  - Business information systems are tied to the appropriate product portfolio subsets, or if your portfolio is sadly static
- No, if…
  - Your product portfolio is dynamic and you try to maintain the same business information systems across all products
- Like accounting, the process is a cycle which you hope continues on a regular basis…

When is it good enough?

- Too many companies suffer from end-user-led perfectionism, and senior-management-led myopia
- An item is “right” when it is “fit for purpose”, i.e. when it meets its true requirements:
  - Input to, and approval of requirements should relate to:
    - Policy / standards – corporate requirements – senior staff
    - Strategy – business unit requirements – business heads
    - Tactics – end-user requirements – end-users

PRO sys

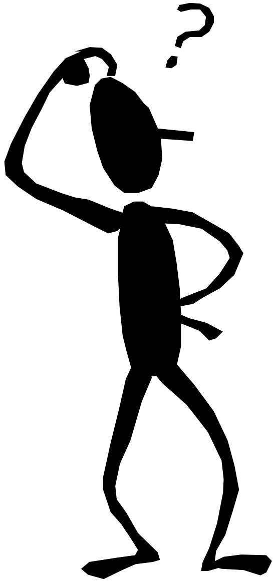
Is the problem ever solved?

- In the LSO world, the problem is touted as an insoluble one
- In other industries, the problem was addressed and solved years ago:
  - Validation = sensible systems definition, selection, and testing
  - Master Validation Plan = Master Business Plan subset
- The one unique problem we have is that because of lengthy product development timescales, regulations and business directions change even while a single product is under development:
  - Hence the need for industry knowledge and a way of building strategy into the plans to cope with changes

After having said all this…

- We purposely made it sound simple
- Reality appears much more complex
- Simple concepts tend to get complex when confronted with content for delivery
  - However: Often reality is complex because the people that made it reality
    - Believe it must be complex
    - Want it to be complex
- Simple concepts, if applied consistently, are powerful tools
  - To keep the solution as simple as possible
  - To maintain focus on what you originally wanted to achieve

Questions, Comments, Feedback?



Jary-Krauser@PROsys-LLC.COM

Achim-Reeb@PROsys-LLC.COM

PRO
sys