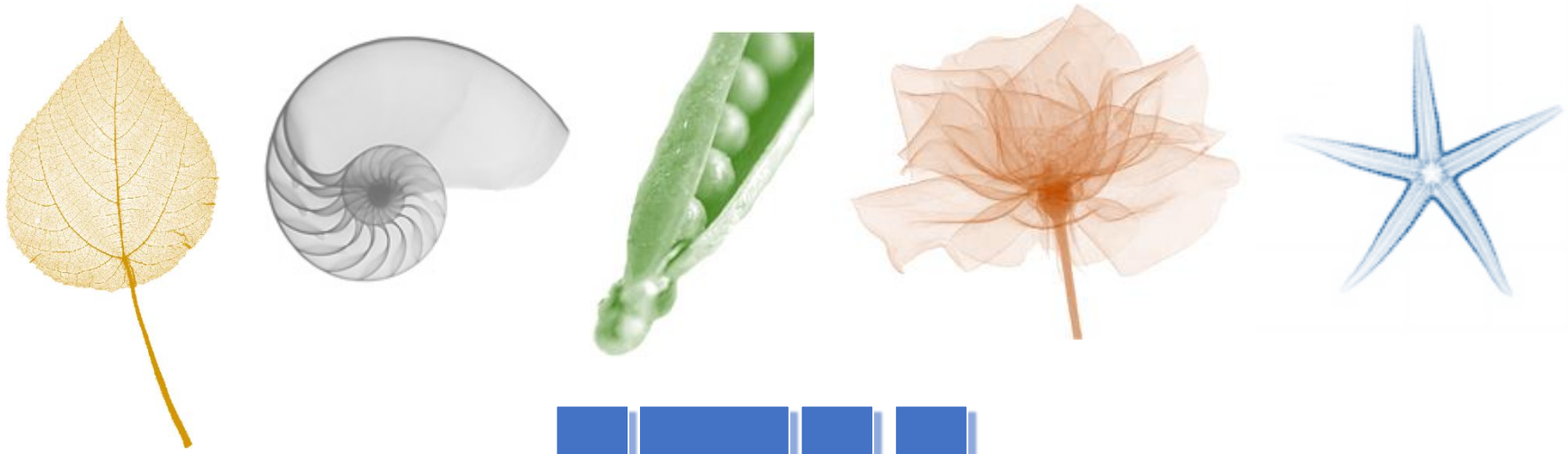


ITIL

A Developer & IT Admin's View

TechSmart Alumni CE – 3/29/2017

Speaker

- Doug Hoff, Centriq Training
  - Java, JavaScript, C#, php
  - business, architecture, programming, analysis, testing, web frameworks

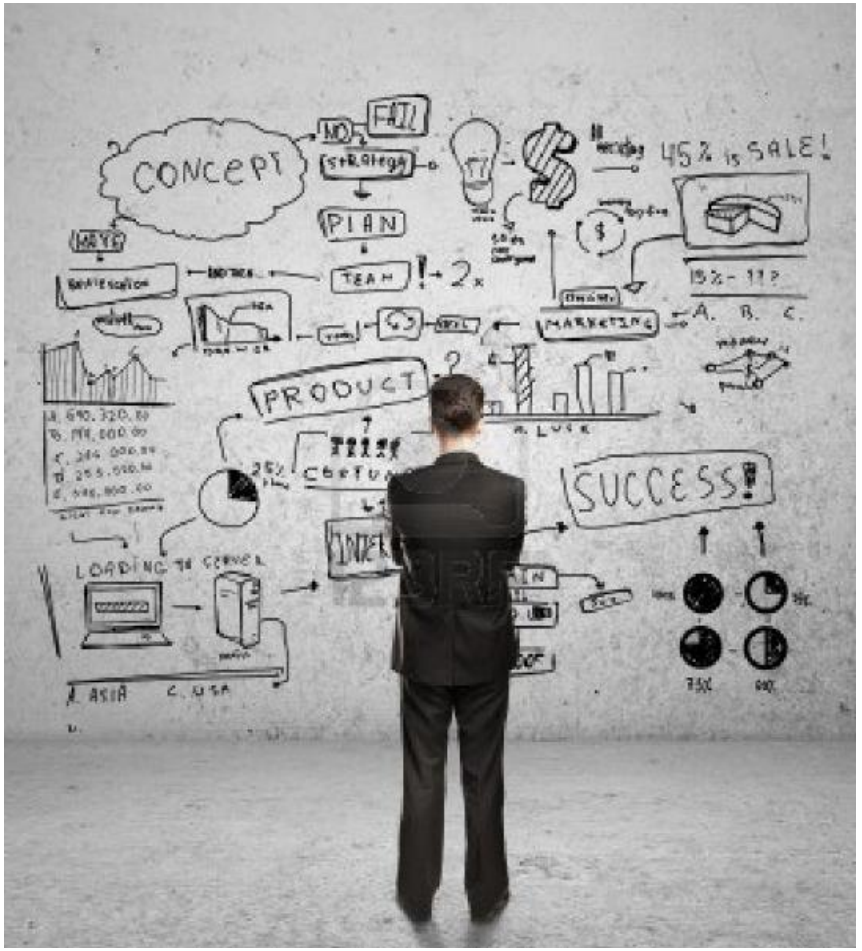
Email	email at doughoff.com
Twitter	@doug_hoff
Web	http://doughoff.com
Blog	http://socialitoutbursts.com
GitHub	https://github.com/doughoff
LinkedIn	https://www.linkedin.com/in/doughoff/

Business and IT

Business service management
IT service management

Why do people run a business?

- A car manufacturer
- A technical school
- A grocery store
- An internet provider
- A clothing store
- A software developer
- A restaurant
- A government



…to provide a service/product

- The customer creates demand
  - by having a problem
- The services create supply
  - to provide solutions
- A business exchanges the service for some value
  - the business model

Business uses IT to increase value

- collect data for reporting
- enforce consistent rules
- eliminate labor costs
- reduce supply costs
- increase communication speed
- create higher quality services/products

Getting more value with IT

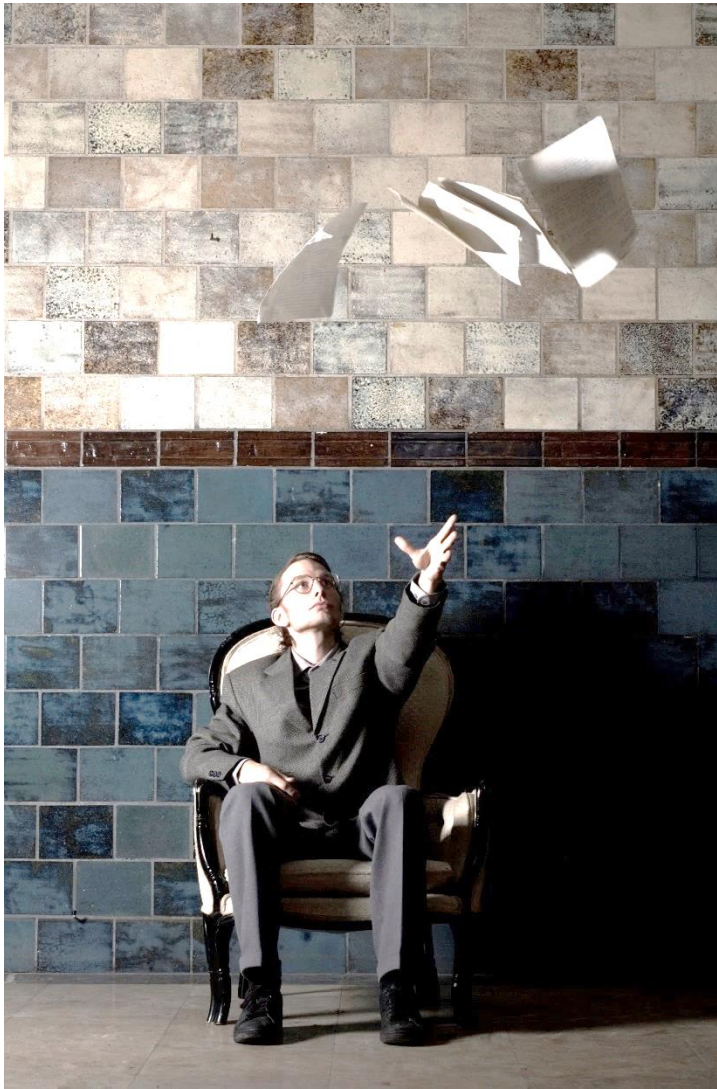
- Use the best IT **service management processes**
  - Measure your current value
  - Recommend **process improvements** for more value
  - Implement the improved version
  - Measure the difference in value

An example of value in management

- Cooking management
  - does it benefit you?
  - who uses it?
  - can you implement cooking?
  - are there multiple best practices of processes?
  - cooking documentation = food science principles
  - processes for me = my cookbook
  - value after one implementation = cooked meal

Another example of value in management

- Gravity management
  - when does it benefit you?
  - can you implement gravity?
  - Would a book “101 processes using gravity” help you?
    - it’s a start
  - Would “How my job should use gravity?” be better?
    - write it yourself, no one else can
    - improve it every day

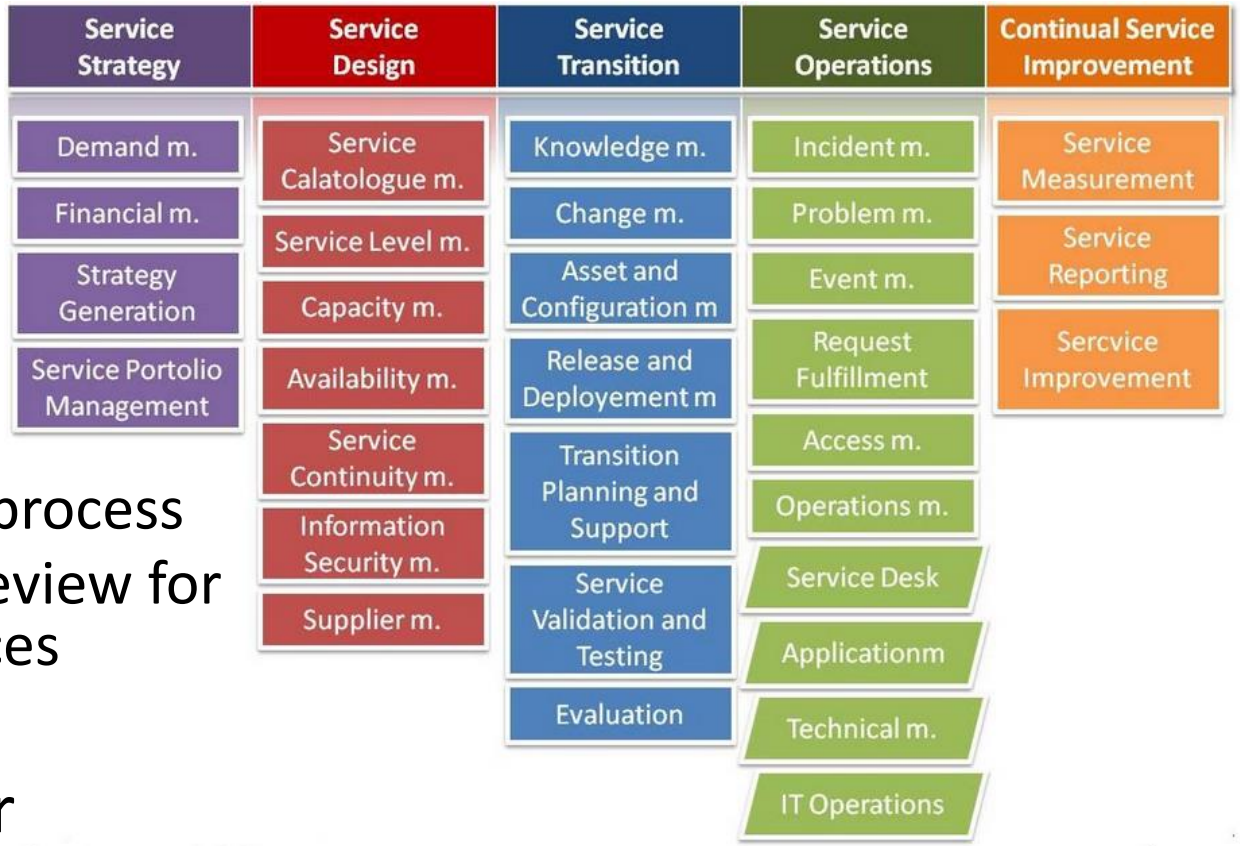


ITIL

- a compendium in 5 volumes about IT and business service management
- processes organized in 4 **project phases**
  - strategy
  - design
  - operation
  - transition
- and one about **quality**

ITIL management processes

- High level business thinking
  - more than project management
  - not an IT development process
  - checklists to review for all best practices
- Use any project management or development workflow

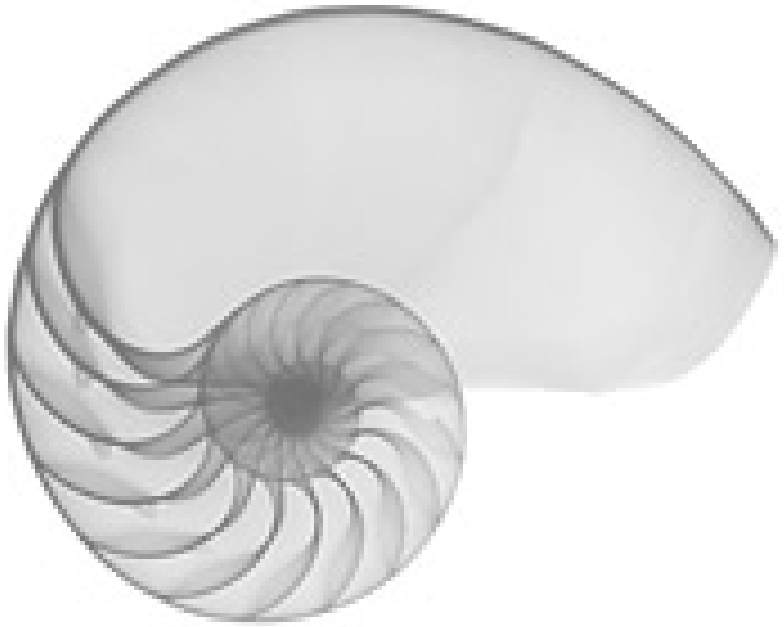


Strategy phase

- Understanding, finding, and maintaining big picture value
- Value comes from services
- Services are anything a customer would pay for
    - products
    - activities
    - guarantees of quality
- Roles
    - executives, upper level managers, enterprise consultants

Design phase

- Planning to add value that you know is there
- Documentation of
  - service development
  - architecture
  - validation of value
  - security
- Roles
  - lower level managers, senior technical staff, business analysts



Transition phase

- Navigating through change to get value
  - keeping risk low
- Maintaining records
- **Build and deploy**
- **Tracking**
- **Testing**
- Roles
  - technical management and staff

Operation phase

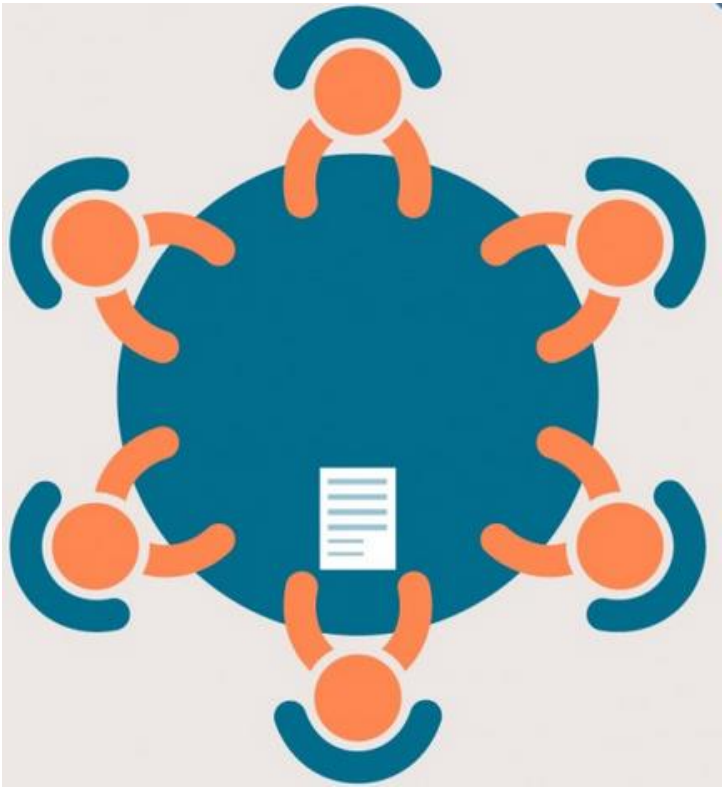
- Getting feedback and ensuring quality of value
- Production
- Service desk
  - incidents
  - problems
  - events
  - security
- Not operations
  (a technical function/dept.)

Continual Service Improvement

- How to improve anything
  - know where you are
  - know where you are going
  - collect data, process it, analyze it, report it
  - correct as necessary
  - know what is success
- Making it “Agile”

Projects need...

- high-level support and leadership
  - the project owner
- requirements
  - service level requirements
  - the service design package
- consistent project management processes
  - the process owners
  - the process managers



Design phase

Requirements gathering, documentation, data and software modeling, technical architecture, security policies

Design – actions

- At initial outset
  - add to total set of services (Service Portfolio) for planning while in design
  - derive capacity from requirements
  - set new budgets now
  - do business impact analysis and risk assessments for disaster recovery, availability and capacity
  - alert service desk
  - start planning development, testing, and deployment
  - set up new suppliers

Design – solution parts

- all business processes
- service itself
  - the requirements
- infrastructure
  - environment for infrastructure
- data
- applications
- support services
- OLAs and contracts – intra-departmental agreements, legal agreements
- support teams
- suppliers

Design – goals

- satisfy business objectives for value
- design for ease and efficiency of development, short maintenance cycles, reduction of long-term costs
- design great processes especially for lifecycle of service
- manage risk early
- design securely and resiliently
- design methods to measure good metrics
- create plans, policies, standards
- enhance skills, capabilities

Design – a balance of…

- functionality (scope)
- resources
  - people
  - technology
  - money
- schedule

Design – four domains

- infrastructure
  - hardware and networks
- environmental
  - physical space, power, cabling, physical security
- data
  - control, access, test data
- applications
  - software, off-the-shelf and in-house

Design – aspects

- the solution itself
  - requirements docs (use cases)
- development tools
  - project management, and full view of all services
  - https://www.thoughtworks.com/radar/tools
- development processes
  - https://www.thoughtworks.com/radar/techniques
- architecture and tools
  - https://www.thoughtworks.com/radar/platforms
  - https://www.thoughtworks.com/radar/languages-and-frameworks
- measurement systems and metrics
  - know what success looks like!
  - correct it early, if it's not in

Design – lifecycle

- Business requirements
  - outline, change request
  - defined – use cases
  - analyzed – prioritized
  - approved
- Customer & support team viewable
  - chartered – resources & budget allocated
  - designed
  - developed / built
  - tested
  - released
  - operational
  - retired

Design – architectures

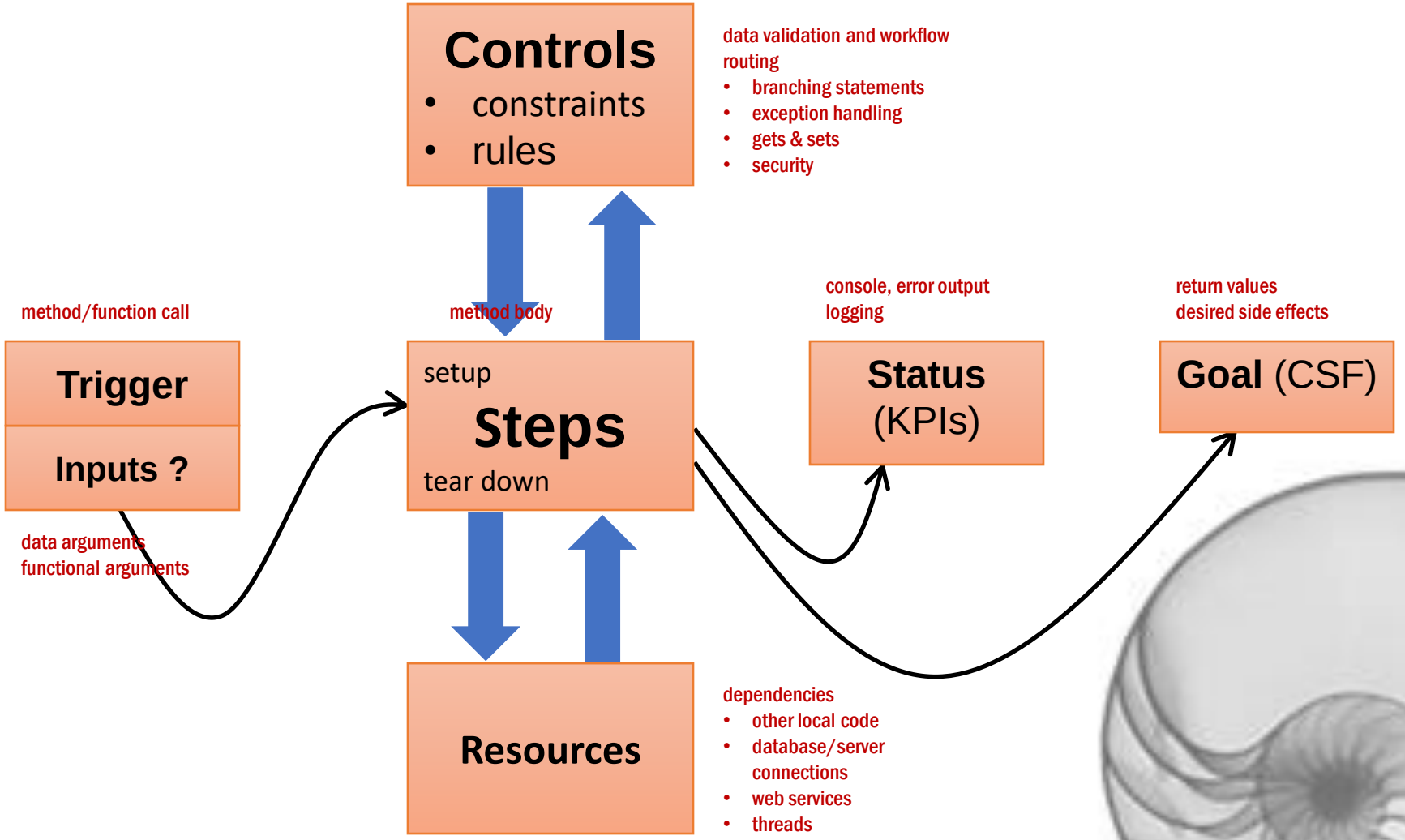
- Service architecture
  - applications, infrastructure, organization, APIs, support → services
- Application architecture
  - business functional needs → applications
- Data architecture
  - logical models, physical data
  - warehouses, datamarts, data lakes
- IT infrastructure architecture
  - hardware/software location, communications
- Environmental architecture
  - environment controls

Design – service catalogs

- the business catalog
  - what does the user see?
  - what do they want to do?
- the technical catalog
  - what are the key functions of the back end?
  - how are they technically managed?
  - mapped to what business service they support

Design – process model

the process parts in computer language



Design - processes

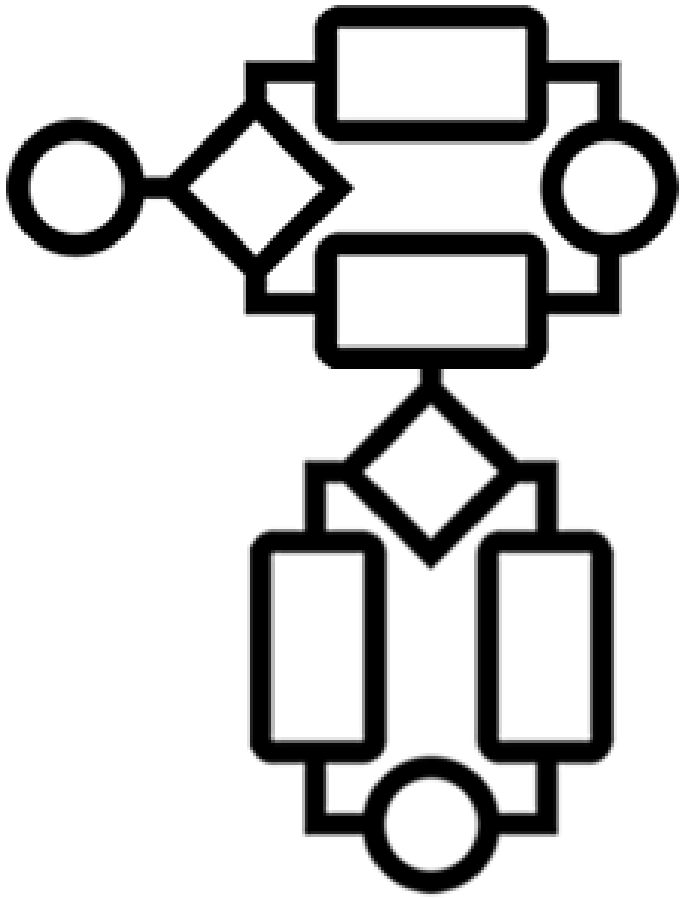
- Updating documentation
  - service catalog management
- Communicating with stakeholders
  - service level management
  - Agile product managers, sales reps
- Understanding scope
  - capacity management
  - availability management
  - IT service continuity management
  - information security management
- Managing external resources
  - supplier management

Transition phase

Development, testing, and release

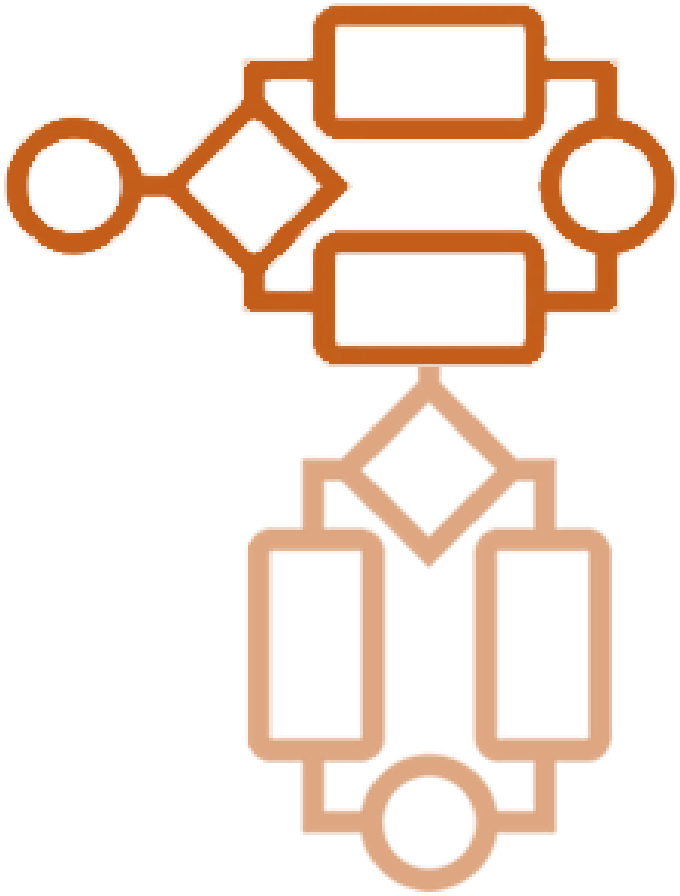
Transition processes - general

- Knowing what you have
  - Service asset management
- Tracking and approving changes
  - Change management
- Knowing what you did
  - Configuration management
- Organizing data
  - Knowledge management



Transition processes – specific

- Project management
  - transition planning and support
- Development
  - release management
- DevOps
  - deployment management
- Testing
  - service testing and validation
- Lessons learned
  - evaluation



Transition principles categories

- Policy
- Change
- Standards
- Re-use
- Business alignment
- Stakeholders
- Controls
- Data management
- Planned releases
- Expect corrections
- Manage resources
- Early warnings
- Quality assurance
- Proactive quality

Transition – policy

- Principles
  - Be clear, align with governance, show commitment by management, integrate teams, address deployment early in design.
- Best practices
  - Transition policy is signed off from management, sponsors, and decision makers.

Transition – change

- Principles
  - Single point of change control, no authority for prod=no access, increase prod environment knowledge, use RFCs, use the ConfigMS
- Best practices
  - Define change, internal and external
  - Justify by business case (value to business)
  - Plan, predict and compare results
  - Management shows commitment
  - CM Audit for unauthorized changes
  - NO acceptance of late changes

Transition – standards

- Principles
  - reusable processes and systems, industry best practices, use ConfigMS, conduct regular reviews and audits
- Best practices
  - publish the standards
  - design a framework for measuring capability and risk profile before and after
  - automate standard processes to reduce adoption resistance
  - derive changes through business case
  - know management commitment
  - encourage buy-in improvement on standards

Transition – re-use

- Principles
  - re-use processes and systems where possible
  - capture data from original sources
  - develop reusable models (flow charts…)
  - use industry standards for better integration
- Best practices
  - use project management practices
  - use existing communication channels
  - follow current HR, training, finance & facilities management practices
  - design transition for customization
  - design transition for consistency with variations

Transition – business alignment

- Principles
  - set user expectations for performance now, let them know how to integrate, check constraints, give users feedback often, monitor and measure in deployment, compare results
- Best practices
  - use project management best practices
  - provide clear and full plans for changes later
  - keep stakeholders involved

Transition – stakeholders

- Principles
  - set expectations for how business will change, communicate changes to all, provide info on how to find transition details
- Best practices
  - verify stakeholders can use it
  - share build, test, and deployment plans and changes
  - work with business level management (executives, marketing) and service level management (sales reps, liaisons) to build customer trust

Transition – controls

- Principles
  - maintain integrity of assets and configurations, automate audits to use more, define accountability and roles at handoff points, use transaction style records for audit trails
- Best practices
  - map roles to processes
  - record role assignments for visibility
  - verify requirements can be done
  - ensure staff making changes is competent
  - make audits

Transition – data management

- Principles
  - right time to right people, train users, improve data, improve business and technical docs, provide easy access, know definitive source of data, show consolidated release data
- Best practices
  - use good presentation and reporting tools
  - vary GUIs by role
  - publish variance reports with risk levels
  - allow access to transition data by service desk, operations, support teams

Transition – planned releases

- Principles
  - everyone is in agreement, coordinated, planned for best use and integrity, risk measured for backing out
- Best practices
  - all updates are recorded
  - definitive versions are stored before test
  - keep a release calendar for planning
  - document prerequisites and co-requisites and communicate to relevant parties

Transition – expect corrections

- Principles
  - communicate that changes will happen, empower that application, learn from the past, use lessons learned sessions
- Best practices
  - use project management for corrections
  - document and control without too much paperwork
  - document changes after baseline
  - involve stakeholders for changes but manage risks and issues internally as appropriate

Transition – manage resources

- Principles
  - recognize resources, skills, and knowledge, assemble teams, use dedicated resources for critical activities, share resources, automate repetitive or error-prone tasks
- Best practices
  - work with HR and supplier management
  - use any competent resource even non-IT
  - proactively handle delays if possible in resource contention
  - measure dedicated vs non-dedicated resources impact on delays

Transition – early warnings

- Principles
  - detect errors as early as possible, prevent resources from being used if wasteful
- Best practices
  - involve customers in design, test planning
  - use testing on how users really use it, not how designers think it's used
  - use past projects for help
  - show value as soon as possible
  - use design audits for risk assessment

Transition – quality assurance

- Principles
  - verify the changes can be made correctly, know what customers really want, test and assess risk thoroughly, make test environment as real as possible, developers and designers **should not** test
- Best practices
  - understand the customer
  - involve the stakeholders
  - know differences between dev, test, and prod environments
  - test environments are under change control
  - start with a baseline

Transition – proactive quality

- Principles
  - detect and resolve incidents early
- Best practices
  - start comparing expectations to results with pilots and early life support
  - do risk evaluation independently
  - use risk evaluation for risk based tests
  - test support tools for messaging/logging feedback
  - encourage transition & operations communication
  - fix errors by priority
  - document your resolutions
  - analyze root causes
  - measure number and impact of incidents per release in test, deploy, and production stages & use to fix underlying problems
  - update incident and problem fix processes

Questions?

Find a pdf at https://github.com/doughoff/business