How FedUni keeps its PeopleSoft world spinning

SESSION 6043
November 7, 2018

BULDING THE FUTURE
ALLIANCE DOWN UNDER

PRESENTER

Nick Balkin

Student & Research Solutions team lead

n.balkin@federation.edu.au

![Federation University Australia logo]

About me…

Started working with PeopleSoft in 2007 on Campus Solutions implementation at FedUni in data conversion.

Moved into development and operational support post go-live.

Involved in bundles and PeopleTools upgrades in 2011-2012.

Appointed technical lead for TAFE implementation in 2013.

Student & Research Solutions Team Lead since 2015.

Tech lead on all PeopleSoft upgrades and major projects

FEDERATION UNIVERSITY AUSTRALIA

- History dates back to 1870 as School of Mines
- Founded in Ballarat, regionally based in Victoria
- Dual-sector
- Approx 23k students

FED UNI & ORACLE

Campus Solutions v9.2, PT8.55
incl. Contributor Relations

Finance v9.0, PT8.53

Oracle 12c

OVERVIEW

1. Operations vs Projects

2. Team structure

3. Demand for change & the Backlog

4. The SDLC & infrastructure

5. Automated environment refreshes

6. Design and development methodologies

7. Operating practices

8. Making life easier
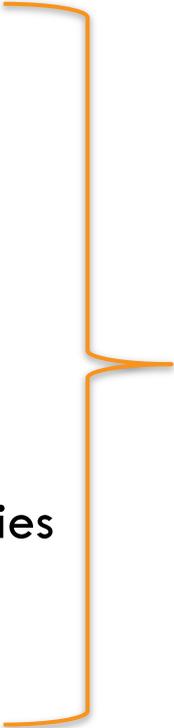
Questions any time!

OPERATIONS V PROJECTS

How FedUni balances the demands of major initiatives while supporting operations

OPERATIONS V PROJECTS

Operations

90% of Operations work is PeopleSoft, and 90% of that is Campus Solutions

1. Lights on
2. Regulation & Legislation changes
3. Enhancement (prioritised from the backlog)
4. Environment management

Projects

- Strategically prioritised investment for 'value-add' delivery

- Isolated resources to ensure delivery

- Upgrades (app and tools), new feature implementations, major enhancements
  - 2018 examples: CS 9.2 upgrade, My eQuals implementation, PeopleTools 8.56, StudyLink, HDR in CS

STRUCTURE

OPERATIONS

PROJECTS

Program Manager

Functional lead

Team lead ⟷ **Technical lead**

Same people!

Senior Test Analyst ⟷ **Test lead**

Hybrid Business/Test Analyst x2

Business Analyst x2

Analyst Programmer x4

Analyst Programmer x1 (+ consultants)

Test Analyst x1

Test Analyst x1

Applications DBA x2

Applications DBA x1

GOVERNANCE

Steering Committee

Operations Group

Core Stakeholder Group

Strategy

Operations
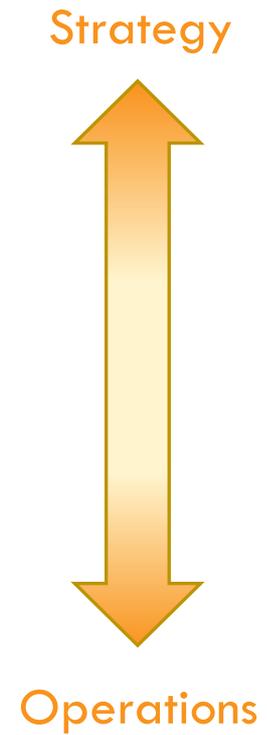
SUPPORT LEVELS

1st
- Servicedesk, ITS

2nd
- Systems & Processes team, Registrars Directorate

**3rd**
- Student & Research Solutions, ITS

THE BACKLOG

- List of Stories (enhancement requests or defect fixes)

- Managed in ServiceNow – Release Management module… with minor customisations

- Story 'State' managed manually

- Stories ranked by Strategic Priority

- Currently around 185 items

- 'Theme' used to group stories

- Detail allows for informative reporting

Strategic Priority	14
Efficiency	3 - High
Recruitment	3 - High
Risk	3 - High
Severity	3 - Normal
Generate Income	2 - Medium
Retention	0 - None

MANAGING THE BACKLOG

- Not everything makes it onto the backlog! Must demonstrate positive return on investment, alignment to strategic objectives, impact and benefit

- Core Stakeholder Group members can escalate

- Resource planning includes standard allocation for BAU support. If BAU support is quiet, resources work on backlog

- Managing backlog size is a challenge

MANAGING THE BACKLOG
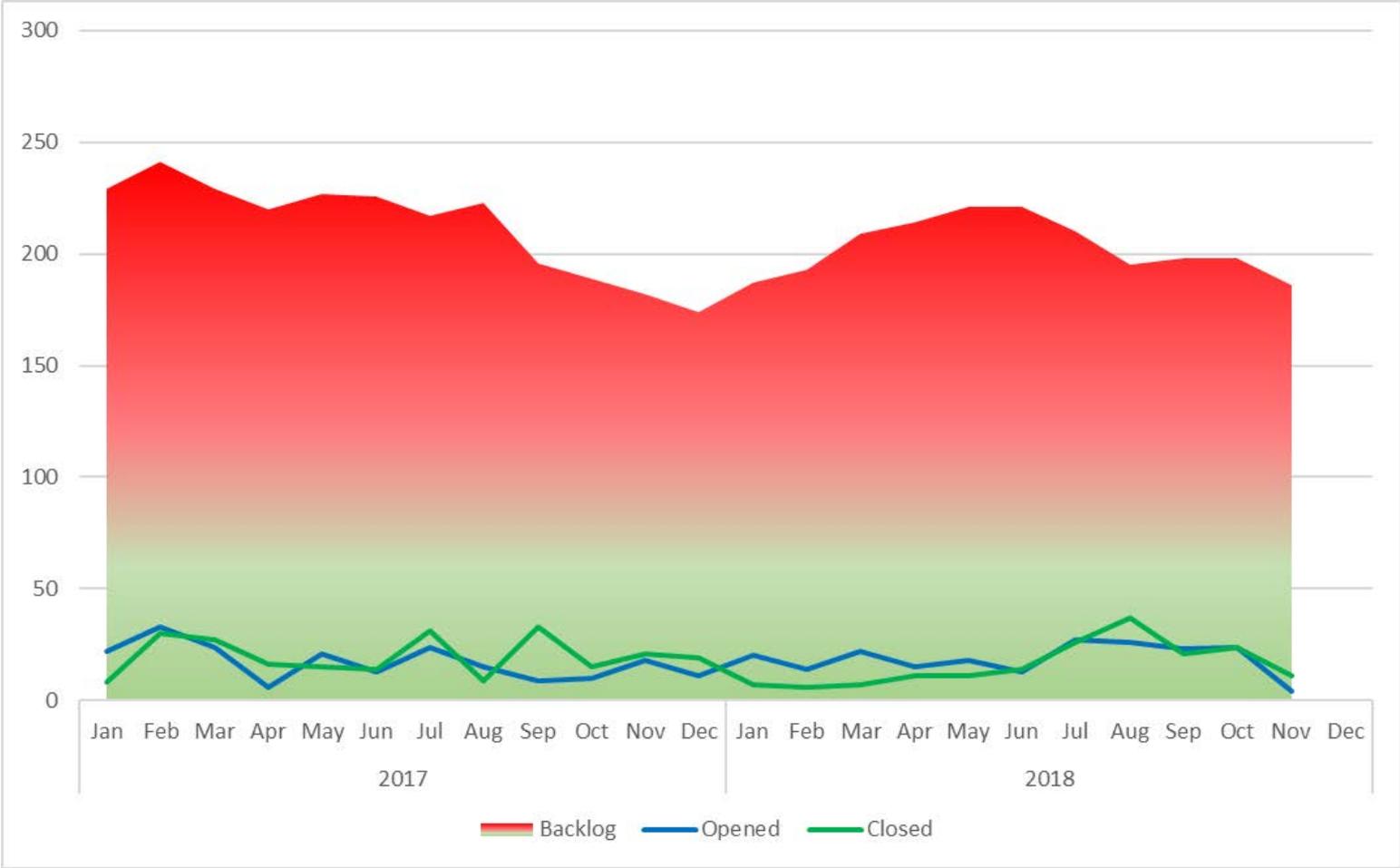
SDLC

Environments, change control, infrastructure, workflow & automated refreshes

SDLC

- 4 environments, one-way flow, single entry point for change

- Clear environment ownership

- Exit gates control release into environments

- Developers do their own migration from DEV to UNIT

- Database System Administrators conduct all migrations beyond UNIT

- Twice yearly refresh from Production

- Manual migration

SDLC - DEV

DEV > UNIT > TEST > UAT

➢ *Primary user: Developers*

➢ *Owner: Team lead*

➢ *Exit control: unrestricted*

➢ *Migrator: Developer*

- Single entry point of change

- Proof of concept work

- Developers can change trace settings and bounce servers

- Solution build

- Developers migrate to Unit test

SDLC - UNIT

DEV > **UNIT** > **TEST** > **UAT**

➢*Primary user: Developers*

➢*Owner: Team lead*

➢*Exit control: Completed technical documentation*

➢*Migrator: Applications Database Administrator*

• Developer ensures solution meets functional requirements

• Technical documentation - migration notes and unit test

• Walkthroughs of development in progress

SDLC - TEST

DEV ❯ **UNIT** ❯ **TEST** ❯ **UAT**

➢ *Primary user: System testers*

➢ *Owner: Test lead*

➢ *Exit control: System test sign-off, completed system test documentation*

➢ *Migrator: Applications Database Administrator*

- Complete test of built solution

- Measured against functional requirements

- Verification of all system functionality

- Broad data scenarios

- Executed by ITS staff

SDLC - UAT

DEV > UNIT > TEST > UAT

➤ *Primary user: Business users, Systems & Processes team*

➤ *Owner: Systems & Processes team*

➤ *Exit control: UAT sign-off, completed UAT documentation*

➤ *Migrator: Applications Database Administrator*

• Managed by central Systems & Processes team

• End to end test according to business process

• Detailed data scenarios

• Performed by requestor, or if unavailable, by Systems & Processes team
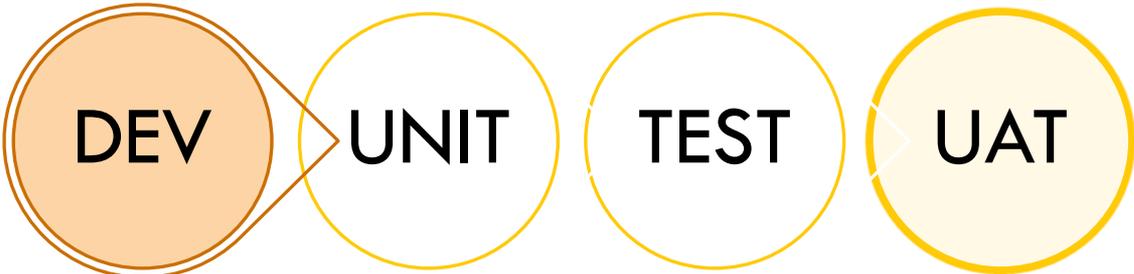
SDLC

CHANGE ADVISORY BOARD

- Comprised of IT Operations, Security and Servicedesk representatives

- Reviews, approves MAJOR change requests before they occur

- Reviews MINOR or PRE-APPROVED changes after release

- Project 'move to Production' reviews
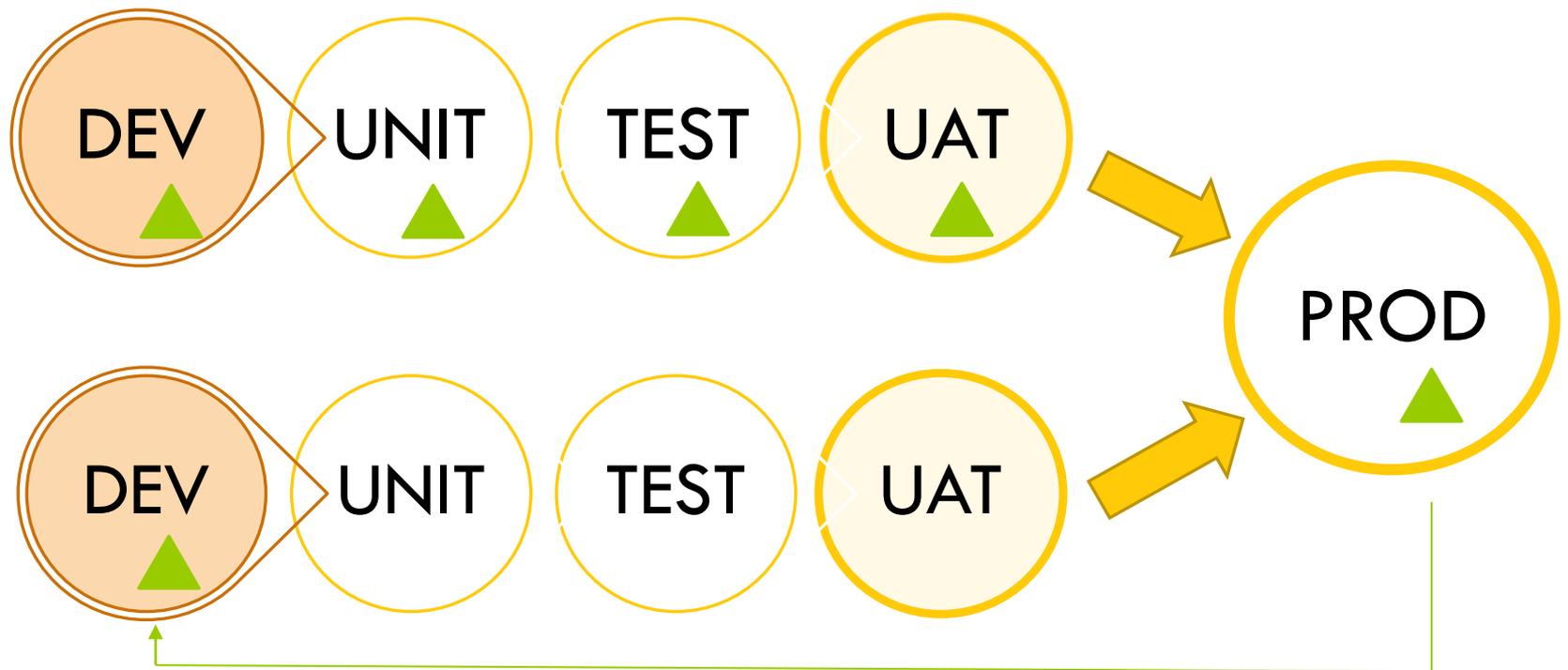
SDLC

ENVIRONMENT STACKS

	SDLC 2	SDLC 1	SUPPORT	PROD
WEB	1x	1x	2x	2x
APP	1x	1x	3x	3x
DB	3x	3x	3x	3x
PSUNX	1x	1x	1x	1x
PSNT	1x	1x	1x	1x

SECONDARY SDLC
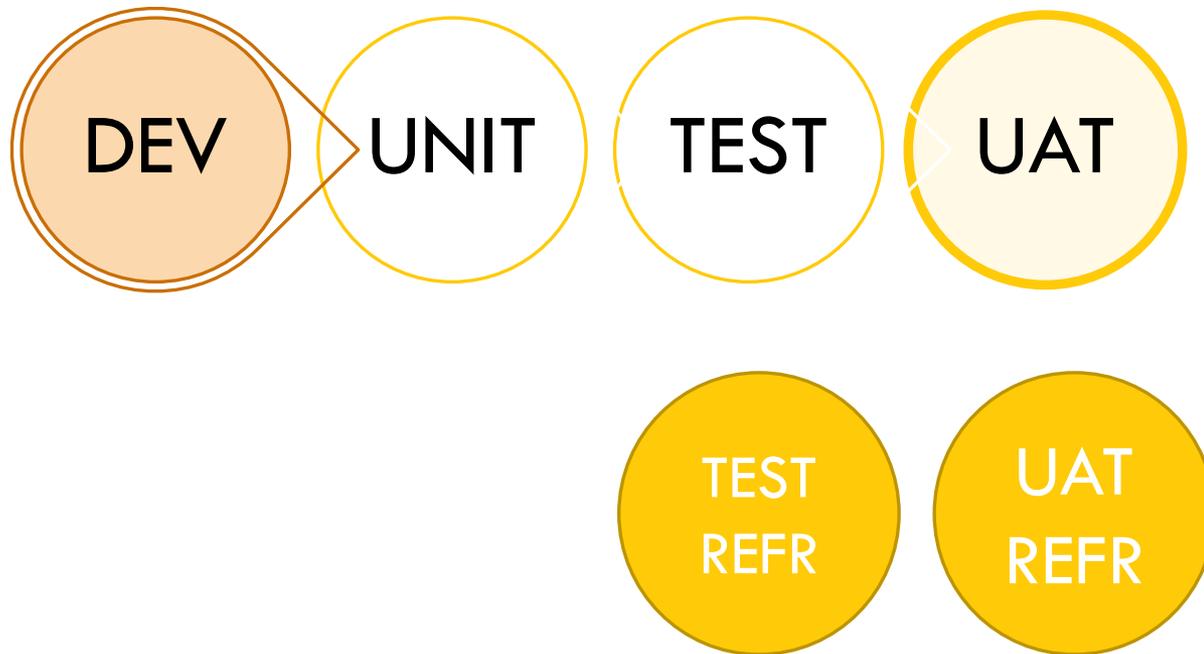
AUTOMATED ENVIRONMENT REFRESH

2x Production replicas for operational support

- Weekly clone
  - 2nd level support,
  - proof-of-concept changes,
  - test-running processes

- Daily clone
  - 3rd level support,
  - troubleshooting data issues,
  - destructive testing,
  - unit and system testing of fixes

Cron based automated refresh
1. Shut down target environment
2. Nightly DB backupset moved to target host
3. Scripts build, localise and 'de-Prod-ify' database
4. Synchronise PS_CUST_HOME changes
5. Start DB, then App, Web, PSUNX servers

DESIGN & DEVELOPMENT

Design ideology,
development methodology

DESIGN IDEOLOGY

- The best flavour is vanilla! We bolt-on...

- Keep it simple
  - Low complexity
  - More flexibility
  - Easier to build
  - Easier to use
  - Simpler to support
  - Easier to take advantage of new features

- Consistency of user experience

- Agreed functional design documents

FUNCTIONAL DESIGN DOCUMENT

Every enhancement must have one!

Business requirements (BR01, BR02)
- Written in business language
- Capture the essence of the problem to be resolved

Functional requirements (BR01_FN01, BR01_FN02)
- Detailed requirements
- One to many Functional Requirements for each Business Requirement
- Written with application terminology & concepts
- The detail that developers code to

Approvers and Reviewers
- Typically 1 to 3 SME's have the final say on approval
- Additional SME's contribute to solution design as reviewers

Includes Security and Testing considerations

Technical addendum (where required)

DEVELOPMENT METHODOLOGIES

Requirements → Design → Build → Test → Release

Agile

Dev | Ops

code, build, test, plan, release, deploy, operate, monitor

Design → Build → Test → Release
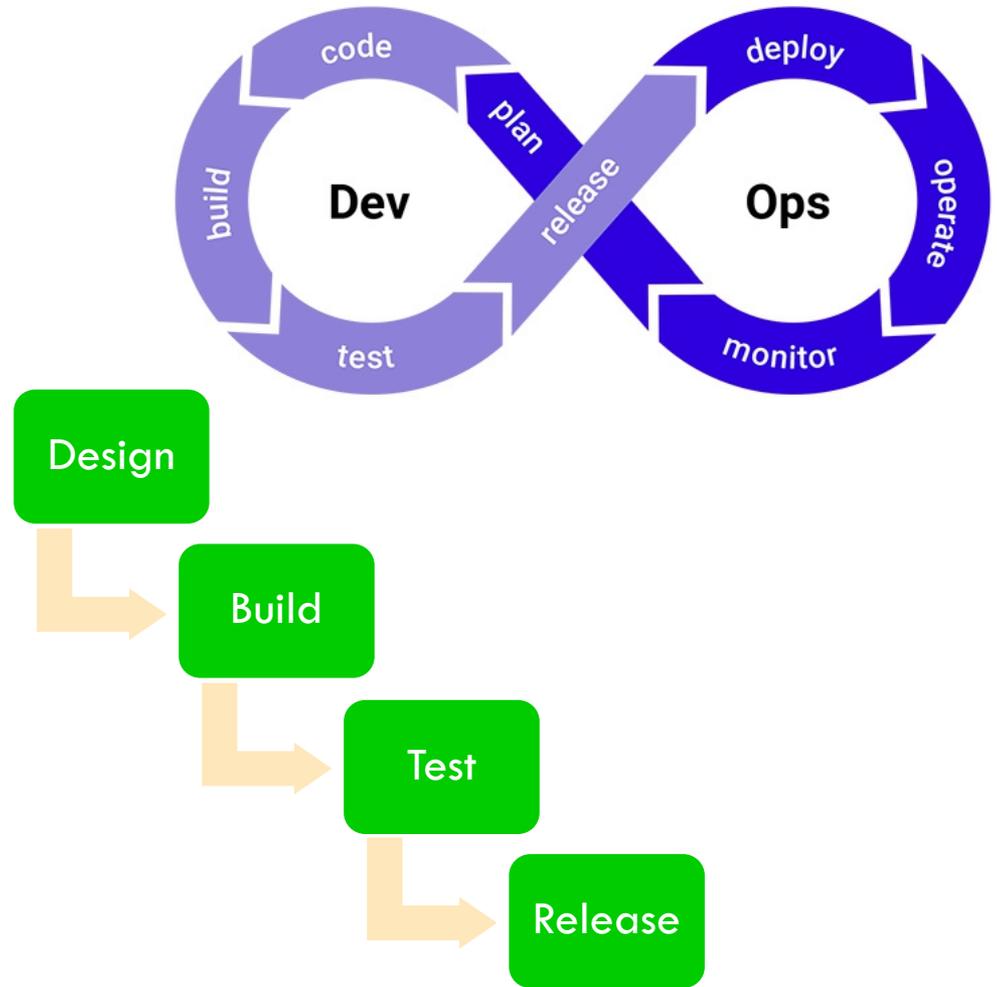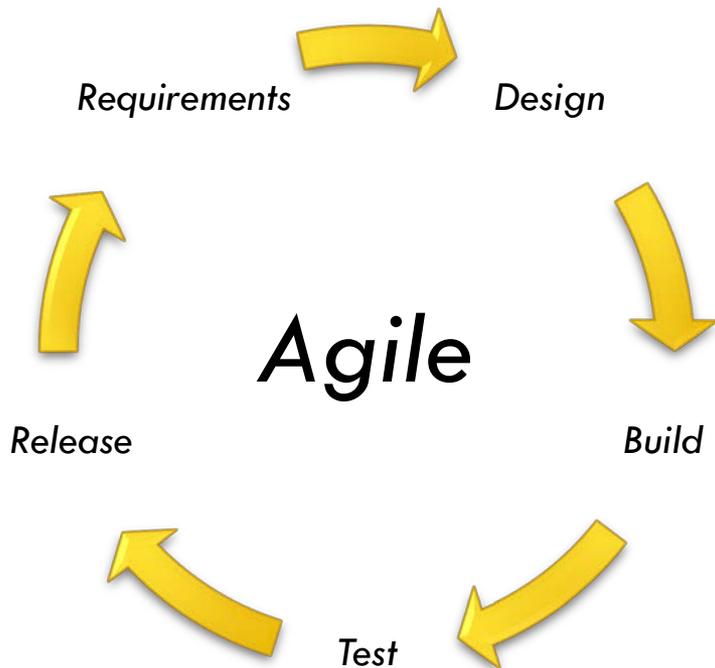
DEVELOPMENT METHODOLOGIES

- Waterfall vs Agile vs DevOps vs…

- We use a little bit of this, a little bit of that

- Monthly prioritisation sessions with core stakeholder group

- Prioritised work follows rigid phases

  Requirements > Design > Development > Test > UAT > Release

- Regular releases

DEVELOPMENT METHODOLOGIES

- Split 'roadblocked' work into smaller chunks and release what we can for earlier return-on-investment

- Responsive to change during design and build.. unless it threatens delivery!

- Projects are generally waterfall

- We avoid Minimum Viable Product

- We don't do 'discovery' well… struggle to prioritise 'innovation' time, to proactively improve

MAKING LIFE EASIER | Concluding thoughts

MAKING LIFE EASIER

Simple, clear rules that are ALWAYS enforced!

- Single entry point for change in sdlc
- No migration into Dev
- No backwards migrations
- Dev's have read/write in dev, read only everywhere else

Automated refreshes

- Daily for replicating Prod issues with like data
- Weekly for proof-of-concept changes, test running processes

MAKING LIFE EASIER

Cross-population of operations staff with ramp-up projects

- Ensures delivery meets internal standards
- Knowledge transfer of business operation to project team
- Knowledge transfer of project activity to operations team

Consistency

- Saves time troubleshooting env
- Hardware
- OS and middleware patch levels
- Application configuration

BLOGS

Use the community!! Blogs from gurus…

support.oracle.com

jjmpsj.blogspot.com – Jim Marion

pe0ples0ft.blogspot.com.au – Sasank Vemana

psadmin.io – Dan Iverson and Kyle Benson

peoplesofttutorial.com – Prashant and Apurva

peoplesoftmods.com – Colton Fisher

peoplesofttipster.com – Duncan Davies

THERE ARE 2 TYPES OF PEOPLE
THOSE WHO UNDERSTAND
MAX(EFFDT)
& THOSE WHO DON'T

QUESTIONS?

PRESENTER

Nick Balkin

Student & Research Solutions team lead

n.balkin@federation.edu.au

THIS PRESENTATION WILL BE AVAILABLE FOR DOWNLOAD FROM HEUG.ORG

SESSION 6043

THANK YOU!

BULDING THE FUTURE
ALLIANCE DOWN UNDER

ANZ HEUG
Australian and New Zealand
Higher Education User Group

HEUG
HIGHER EDUCATION USER GROUP ™

ADU 7-9 NOVEMBER 2018