



SWIN  
BUR  
\* NE \*

SWINBURNE  
UNIVERSITY OF  
TECHNOLOGY

# The Future of Software Engineering in Australia

Prof John Grundy FASE FIEAust  
Dean, Software & Electrical Engineering

Swinburne  
►think forward

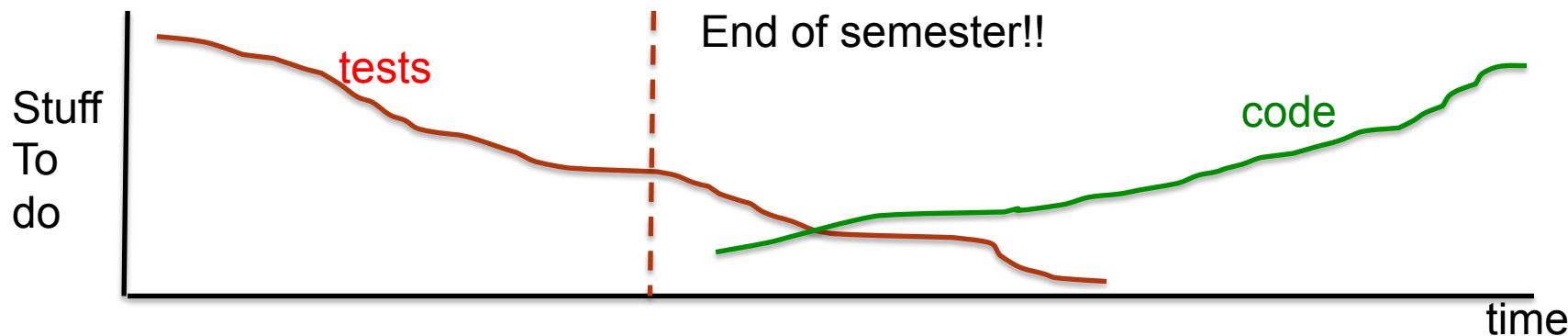
## Outline

- Motivation: some examples & key lessons
- SE Research in Australia
  - Some of our work to address
- SE Teaching in Australia
  - Some of our work to address
- SE Practice in Australia
  - Some of our work to address
- Where to from here?



## Need for SE – Example #1

- Two excellent final year BE(Software) students & their capstone team project
- Personal health care planning app for mobile (this was mid-2000's!)
- Totally sold on concept of Agile and heavily adopted Test-first development approach...



[ Note Phillip Kruchen's observations on refactoring-out-of-control!! ]

## Need for SE – Example #2

- Consulting for start-up developing in-memory database technologies
- Migrate business systems to using innovative approach to process automation, data analytics, dashboards etc [ this was year 2000/1 !]
- Nicely speced, prototyped, demonstrated novel system intergation / in-memory database-based product
- Desktop-based client ; shared enterprise server
- No take up / no further development – clients wanted web-based ERPs, functionality vs speed, ACID vs non-ACID transactions, ...

[ I note many large organisations e.g. SAP now have such product suites for exact same market... ]

## Need for SE – Example #3

- Built a sophisticated ERP system
- Delivered to range of local government users
- Got rung up about major database corruption problem
- After discussion needed to do full restore from backup
- But...
  - Daily tape corrupted
  - Weekly tape = daily tape (!)
  - Monthly tape = weekly = daily tape (!!!)

## Lessons

- Example #1 – the students & their chosen approach
  - Do it right – choose right technology, process ; do it accurately, on-time, on-budget, etc
- Example #2 – the in-memory database product
  - Do the right thing – solution didn't meet customer needs / requirements
- Example #3 – the client's (lack of) backup system
  - End users need to do right/do right thing too – real people use real systems – user do needs to be right/right thing too!

So how do we ensure these are learned from?

- SE Research – theoretical/conceptual, experimental, applied
- SE Education – by itself and with other disciplines
- SE Practice – engagement with practitioners, learn from practice
- I'll briefly review these, their Australian context, some research we are doing to (try and) address aspects of each...

## SE Research in Australia

- The ERA2012 round and attempts to propose an SE-oriented Centre of Excellent suggest SE research is relatively weak in Australia
- Many unis have no or maybe 1 person doing SE research (and publishing in high repute venues like TSE, ICSE etc)
- Swinburne has by far the largest concentration ☺
- Given industry demand for SE skills – why is this? Does it matter?

## SE Research in Australia

- I think it does
- We limit our ability to contribute to SE research worldwide
- We limit our ability to work more closely with industry
- We limit our ability to provide students – CS and not just SE – with leading-edge research outcomes
- We limit our ability to inform our curriculum and engagement with these outcomes
- We limit our ability to leverage SE expertise in other areas

## Why this limited research effort

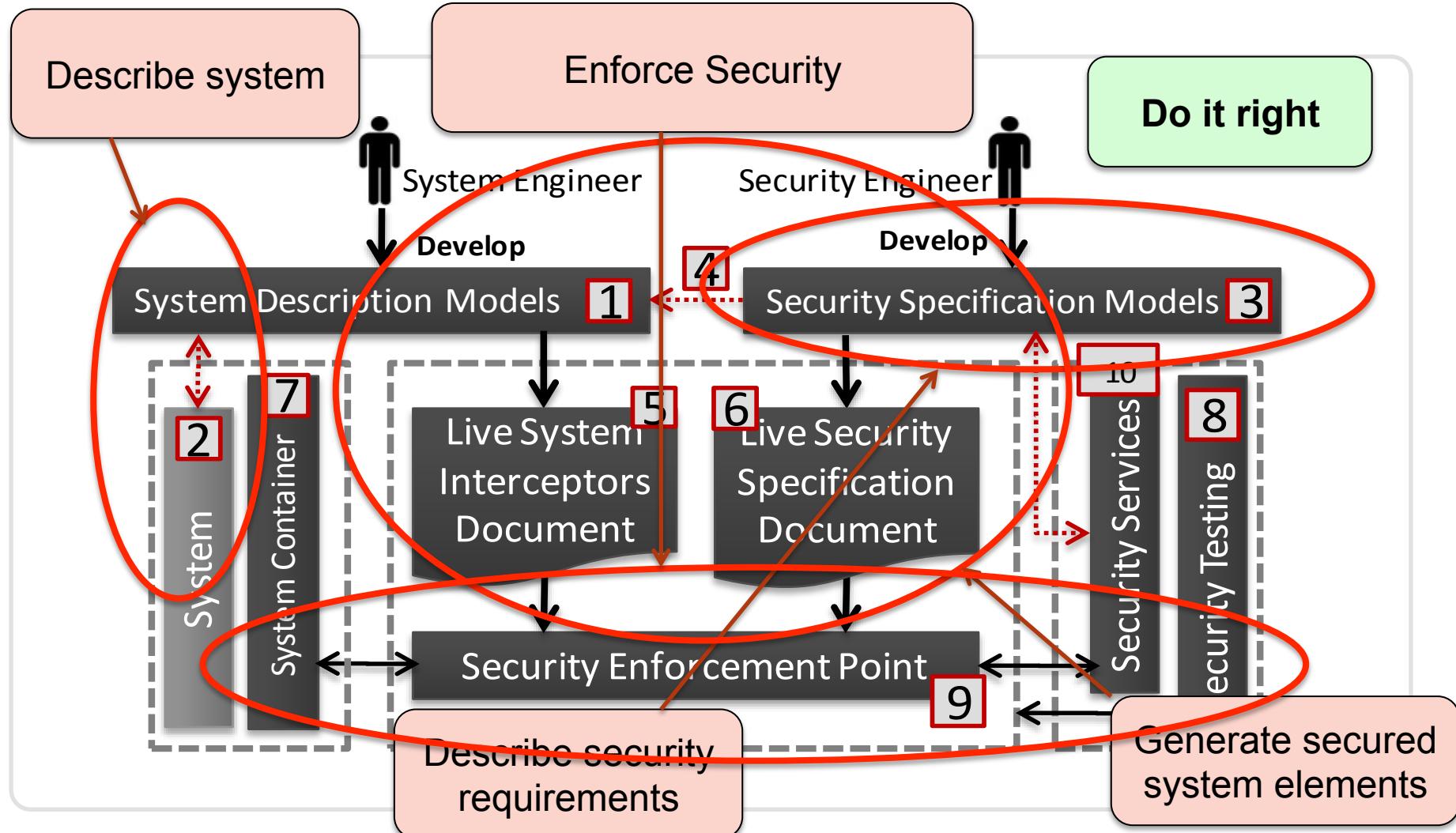
- Much focused on Computer Science technology or theory contributions e.g. ERA 0801 (AI & Vision) very strong ; ERA 0806 (“Information Systems”) over 30 Units of Evaluation
- Some overlap e.g. SE & software applications in general & distributed systems
- We don’t always work as well as we should with other areas – in CS and more broadly
- SE student numbers (I’ll come back to this) – where most \$ comes from – dramatically fallen vs ICT in general

## Some areas NEEDING solid SE contributions

- (Software) Security engineering
- Mobile device applications
- Information Visualization applications
- “Big data” management and analytics
- End user computing

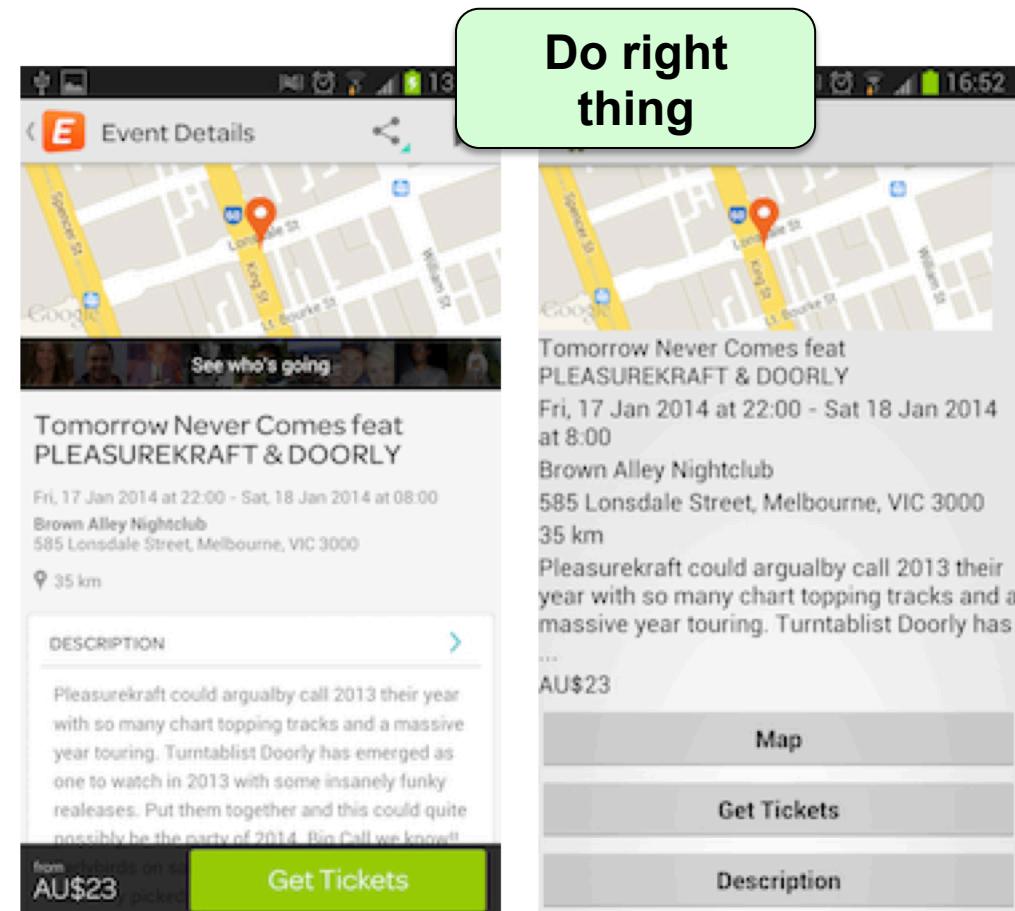
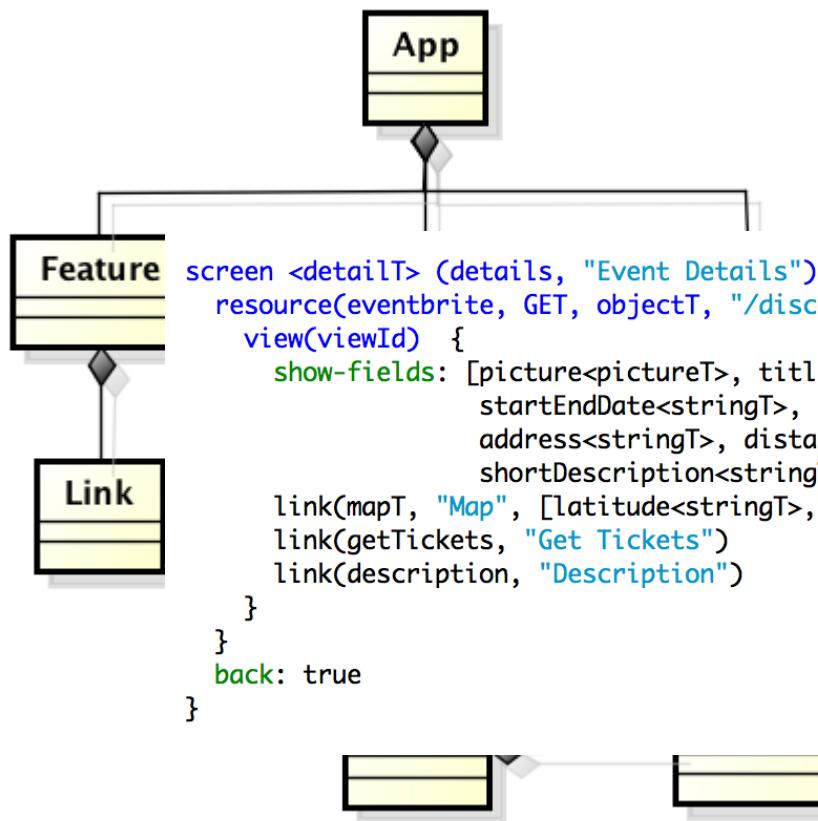


## Example #1 - Model-driven Security Engineering @ Run-time





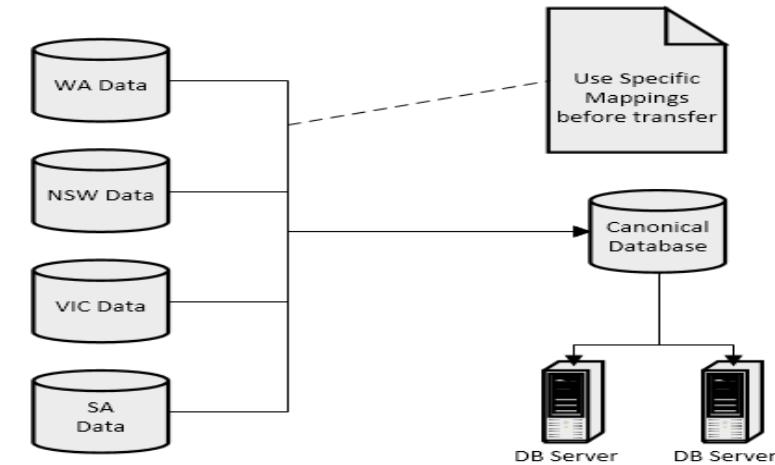
## Example #2 – Generating mobile app prototypes





## Example #3 – Big Data visualisaiton - Example

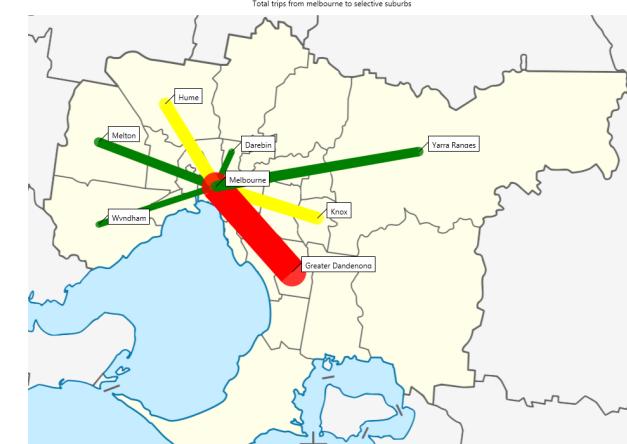
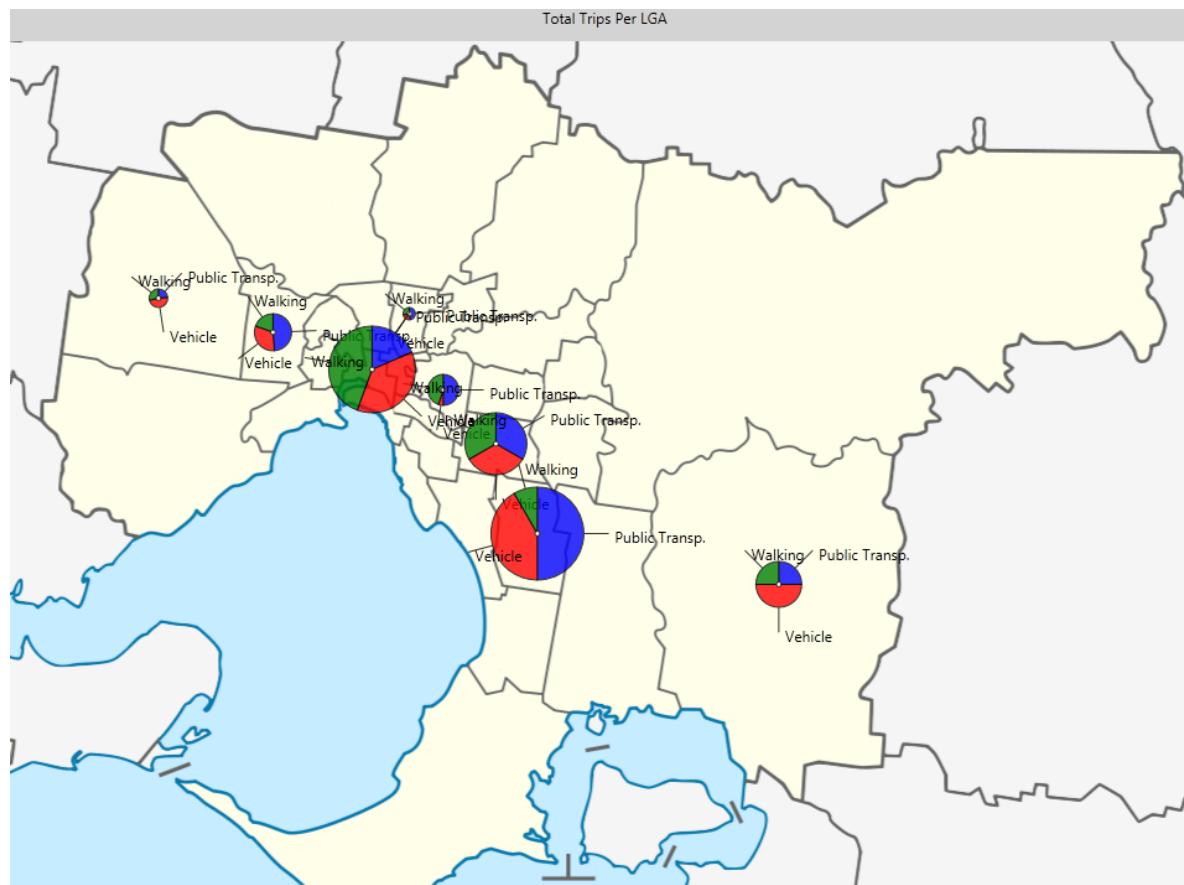
- Various states with different Survey instruments and categorisation
- Harmonise into a canonical DB
- Project includes data wrangling, mapping, and visual analytics



Travel purpose inconsistency samples				
VIC	NSW	Harmonised	NSW AGG	WA
N/A Missing		Missing		Default N/A Missing
Buy Something	Shopping	Buy Something	Shopping	Buy Something
Education	Education Childcare	Education	Education_ChildCare	Education Childcare
Work Related	Work related business Return to other job Return to main job Go to other job Go to main job	Work Related	Work Related Business	Work Purposes
Social	Social welfare Social visits	Social	Social_Recreation	Social
Recreational	Recreation	Recreational		Recreational



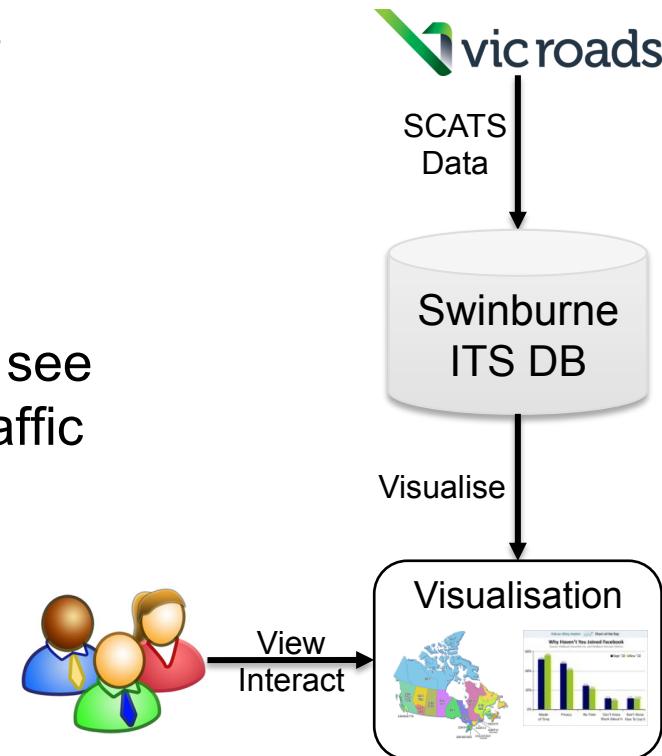
## Sample Visualisations of Harmonised Data





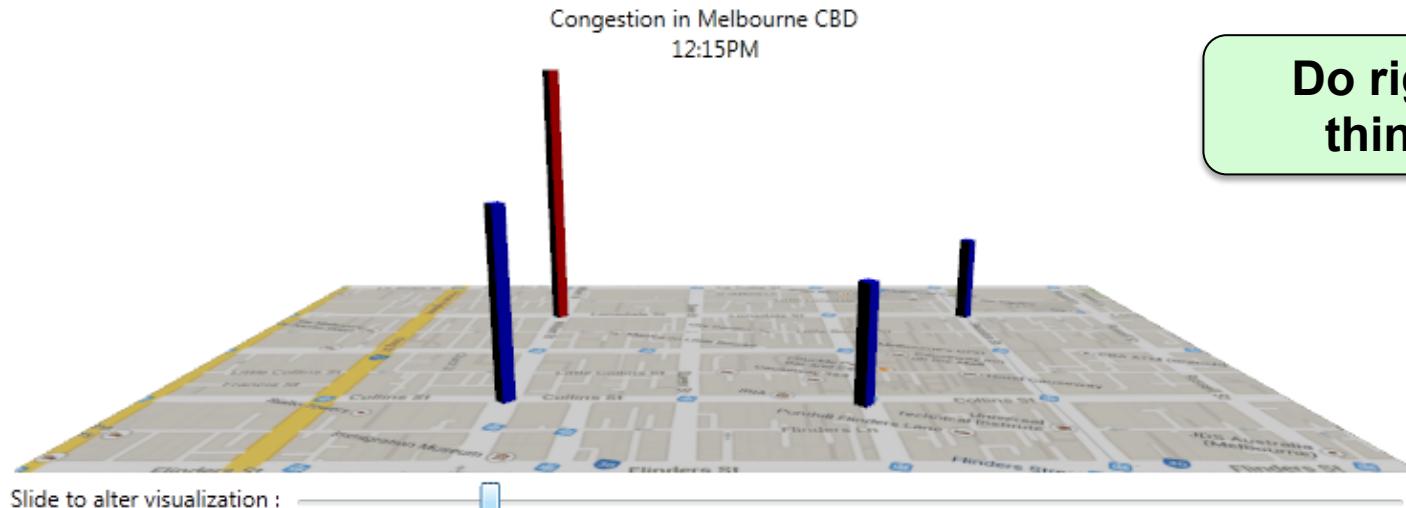
## Big Data Visualisation for Traffic Management

- VicRoads Collects traffic data (SCATS)
- Data is transferred to Swinburne's local DB daily.
- Project:
  - Visualisation for collected data.
  - Users can interact with visualisations to see different aspects of the data and help traffic management.

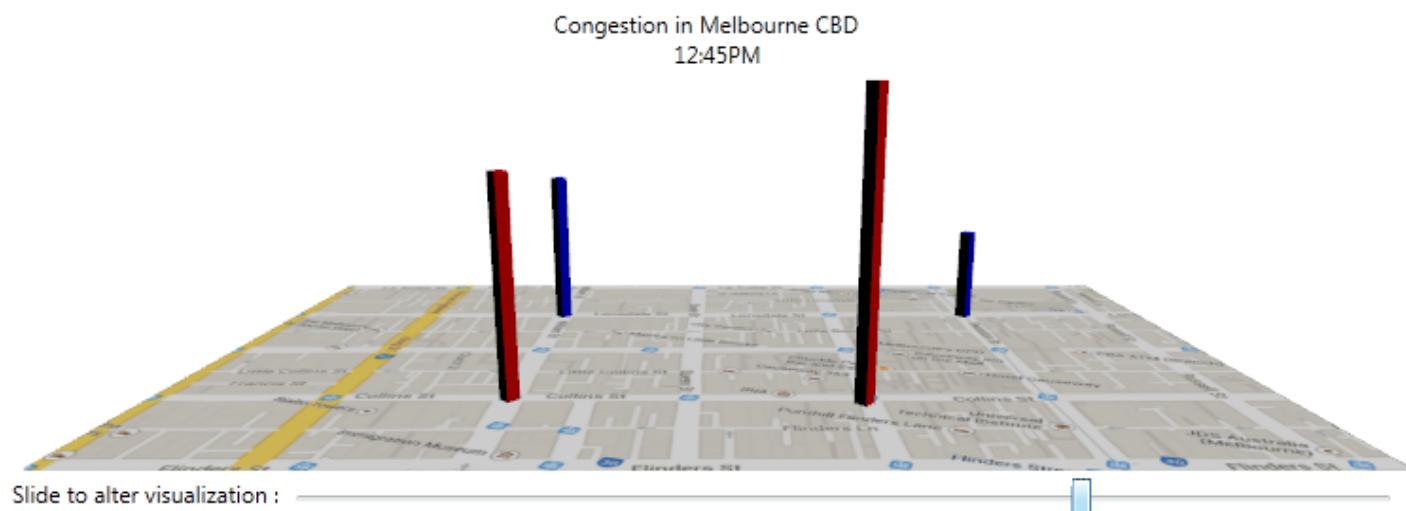




## Sample Visualisations Traffic Data



Do right  
thing



## SE Education in Australia

- Late 90s, early 2000s – many new BE(SE) degrees introduced / many CS degrees enhanced with SE content
- Last 5-10 years
  - Many BE(SE) degrees discontinued (Swinburne one “re-booted” in 2012...)
  - Many CS degrees introduced new content but (relatively) little SE focus
  - Demand for SE expertise in industry has significantly increased e.g. @ University of Auckland – producing 30-60 BE(SE) grads a year but (way) more demanded

## Some SE education contributions needed

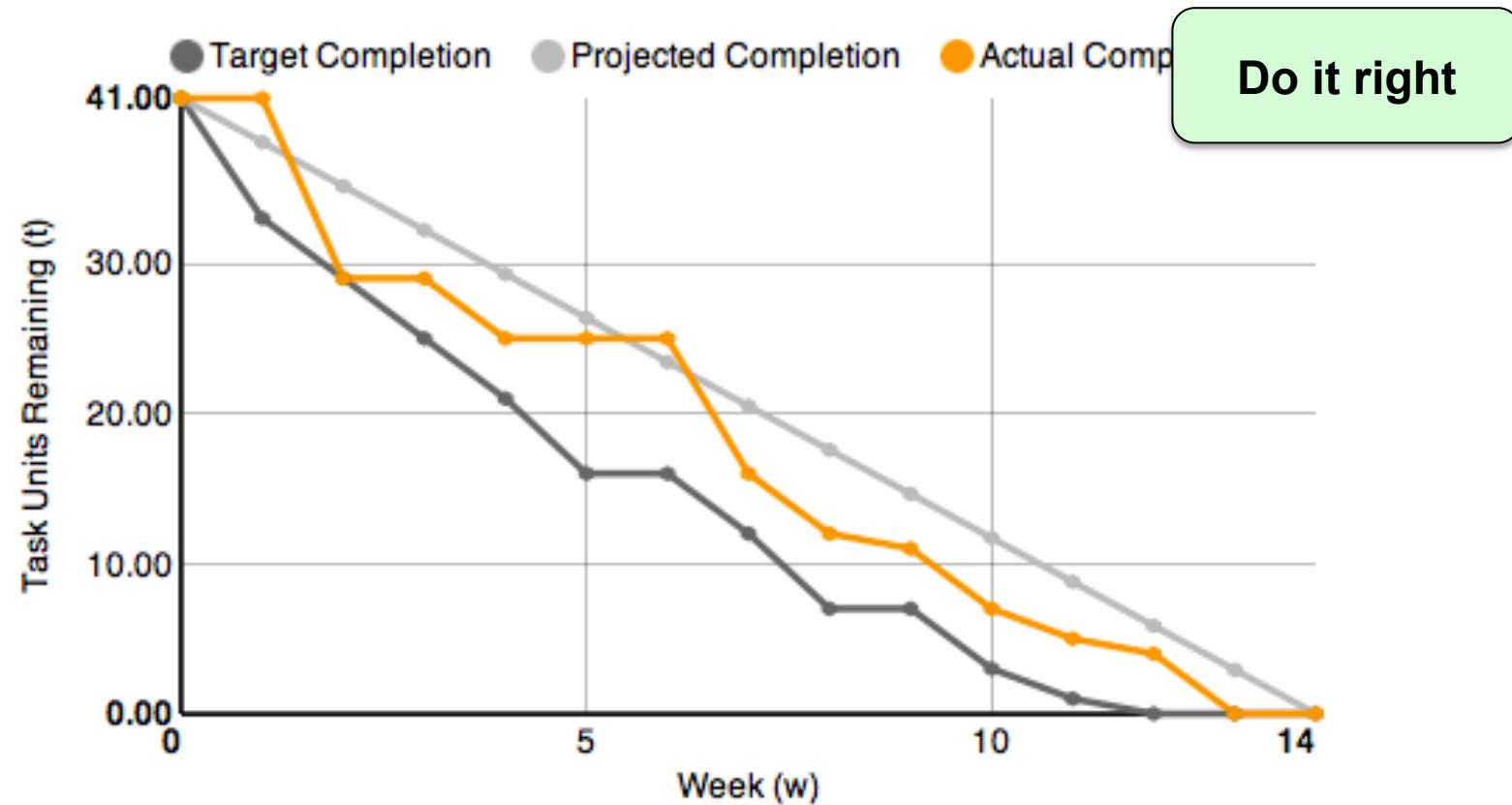
- Processes e.g. Agile development methods
- More “authentic” and engaging assessment approaches e.g. portfolio, peer review
- Industry-readiness via capstones and other projects e.g. industry-based projects, internships
- Incorporating new and emerging technologies / approaches: mobile apps, security engineering, cloud computing, end user computing, big data analytics, sensor networks and ubiquitous computing systems
- Exposing more students to SE theories, practices during their programmes

## Example #1 – Constructive Alignment & Portfolio Assessment

- Andrew Cain's PhD
- Idea of the constructively aligned curriculum
- Portfolio-based assessment practices
- Initially applied to programming units
- Applying to other SE units e.g. mobile app development, games programming, software deployment & evolution
- Improved engagement, pass rates, retention



## Example: Doubtfire tool

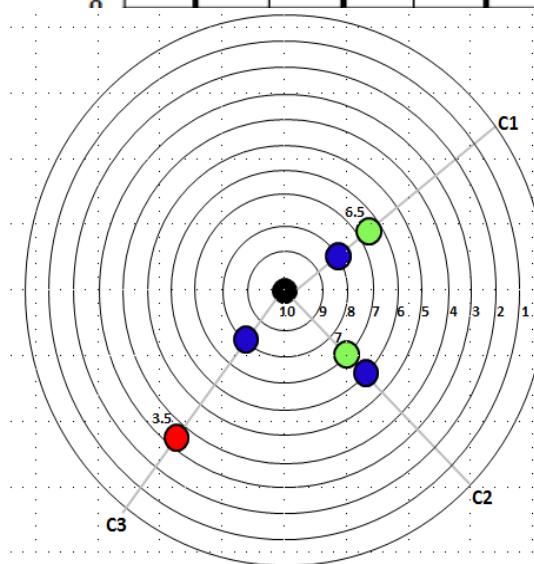
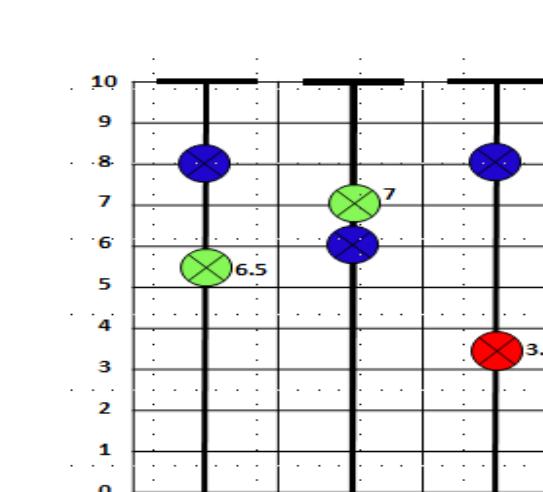
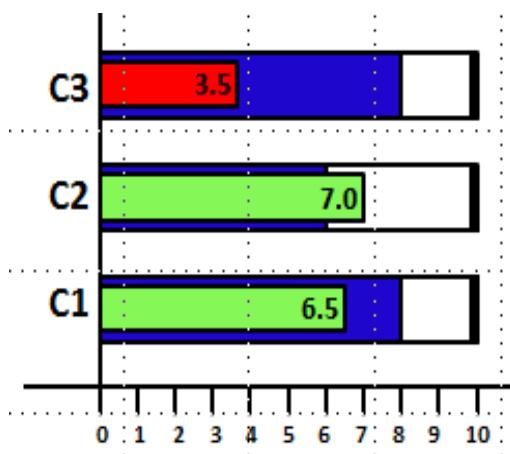
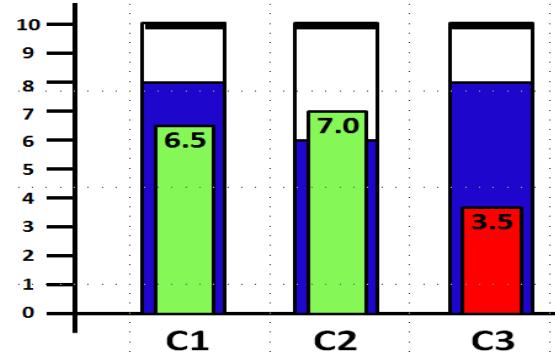


## Example #2 –Open Learner Model/ePortfolios

- Check Law's PhD
- Idea of surfacing competencies / learning outcomes and progress towards
- Idea of evidencing via ePortfolios
- Idea of charting individual, team, class progress ; proactive interventions ; visualisation of progress metrics
- Applying to programming, mathematics, design, deployment & evolution units



## Example – progress against Engineering competencies



## SE practice in Australia

- Many SMEs ; arguably most innovation/potential in this space
- More and more SE need from non-SE/ICT companies
- We do pretty well – but how do we do BETTER?
- How can we better innovate with Software Engineering approaches, tools, methods etc?
- How do we leverage emerging areas of big data, cloud computing, mobility in our application engineering?
- How do we more effectively work together (SE researchers, practitioners, students)?

## Example #1 – What do testers do? How improve?

- Studying what software testers in industry do ; Do it right improve development of testers
- Work log analysis
- Job descriptions
- Tasks and time spent on tasks
- Personality impact on SE testing – what makes a good software tester?
- How to integrate into SE testing in our units ; how to integrate into SE testing practices in industry...

## Example #2 – Software Requirements & Architecture

- Studying current SE requirements and architecture practices
- Requirements capture techniques
- Requirements engineering processes
- Improved tools for RE
- Software architecture documentation approaches
- Software architecture design, analysis and improvement



## Example: GUITAR capture & analysis

Do right thing

The screenshot illustrates the GUITAR tool's workflow for capturing and analyzing artifacts:

- Left Panel (Artifact Capture):** Shows a tree view of artifacts and a form to "Add New Artefact". The form includes fields for Artefact Type (Functional Service Goal), Artefact ID (FSG30), Text Description (<Users> shall be able to <...>), and Structured Description (Agent(Head(User)) + Verb(Head(Review)) + Qualifier(Activity('CreateReview'))). A red annotation "structured specification" points to the structured description area.
- Middle Panel (Analysis):** Displays a "Quick Fix" dialog. It lists potential fixes for detected inconsistencies, such as "Delete artefact FSG30" and "Delete restriction in the description of artefact FSG30". A red annotation "a list of resolution alternatives" points to this list. Below the fixes, a "Problems" section shows a single problem entry: "Resource: SocialNetwork Location: Unknown". A red annotation "highlight problematic artifacts" points to the "SocialNetwork" entry.
- Right Panel (Graph View):** A graph visualization showing relationships between artifacts FSG5, FSG30, FSG6, and FSG4. FSG30 is marked with a red X, indicating it is problematic. FSG6 has a red X above it, and FSG4 has a red X below it, both pointing to the "highlight problematic artifacts" annotation.
- Bottom Panel (Problem Description):** A table titled "Inconsistency among artefacts are detected" with rows for "FSG30, FSG6" and "Logical Inconsistency". A red annotation "Problem description" points to this table.



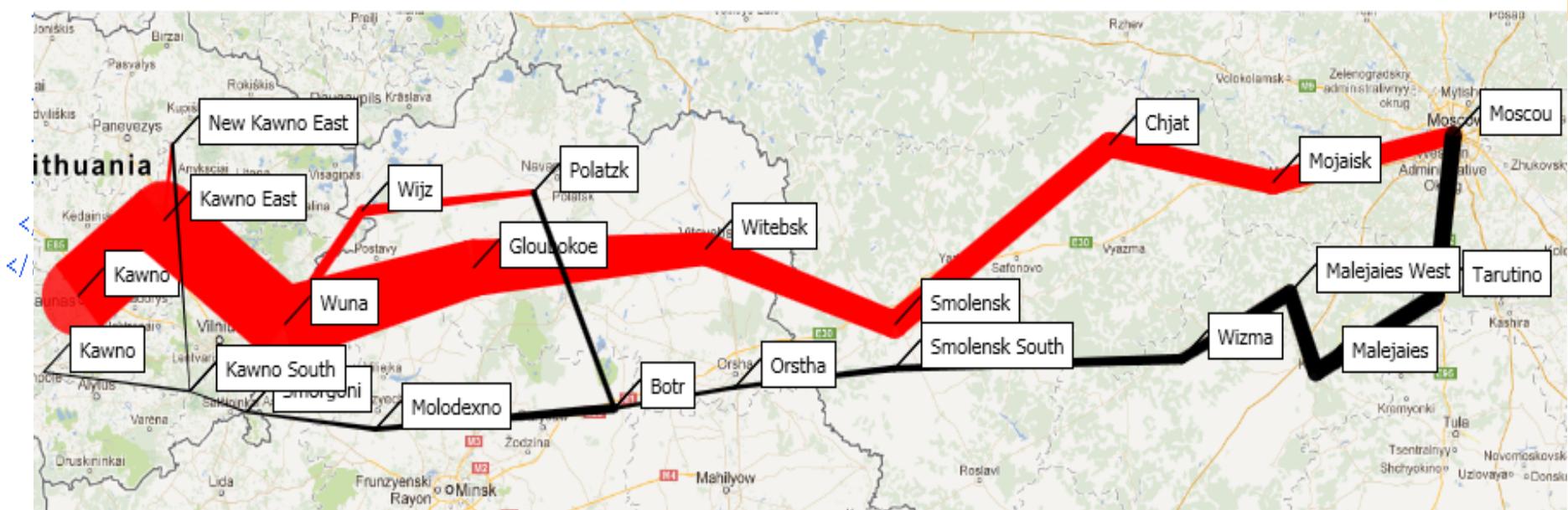
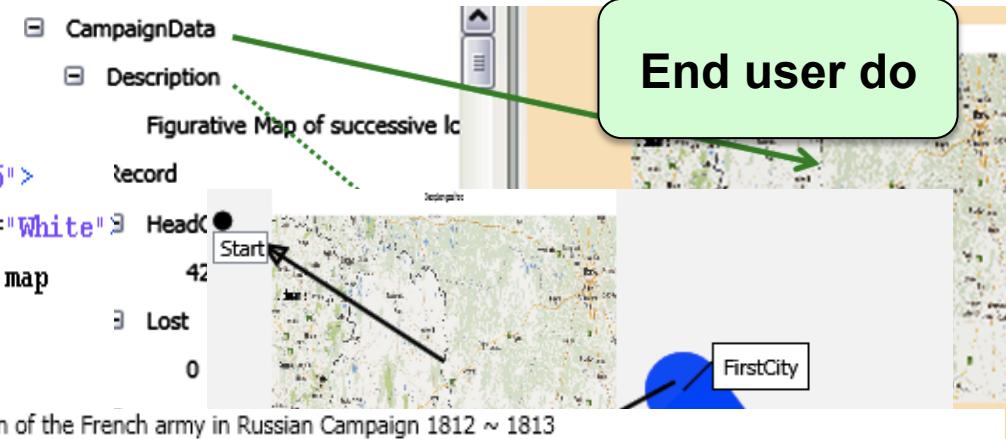
## Example #3: Generating InfoVis ... End User “SE” support

```

<MapData>
  <Description>Description goes here</Description>
  <Movements>Troop Movements</Movements>
</MapData>

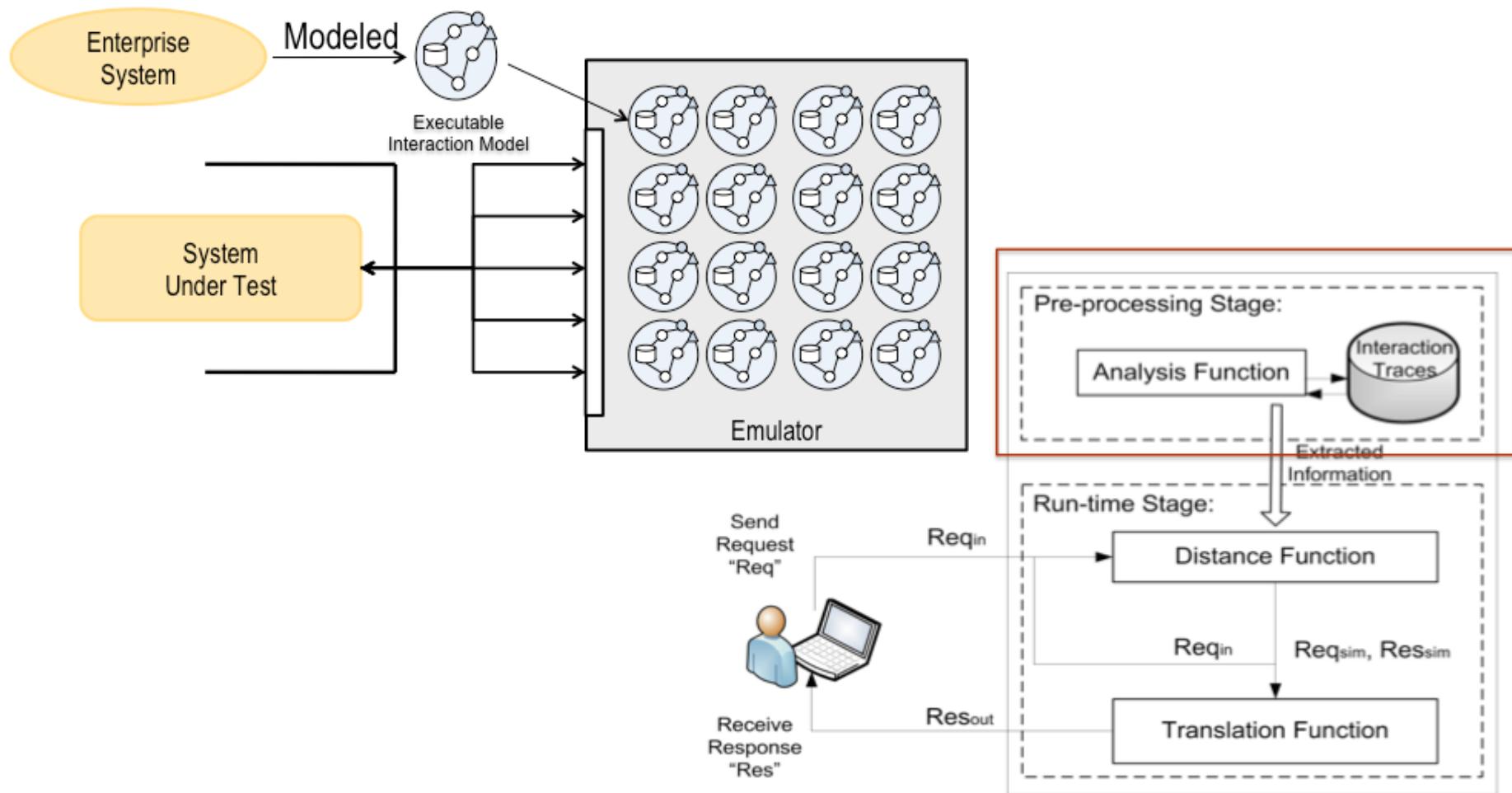
<StackPanel Orientation="Vertical" Height="290" Width="716">
  <TextBlock Height="27" TextAlignment="Center" Background="White">
    <TextBlock.Text linkto="Desc">
      FirstCity
    </TextBlock.Text>
  </TextBlock>
  <TextBlock>
    of the map
    SecondCity
  </TextBlock>
</StackPanel>

```



## Example #4 - Enterprise Emulation Tool

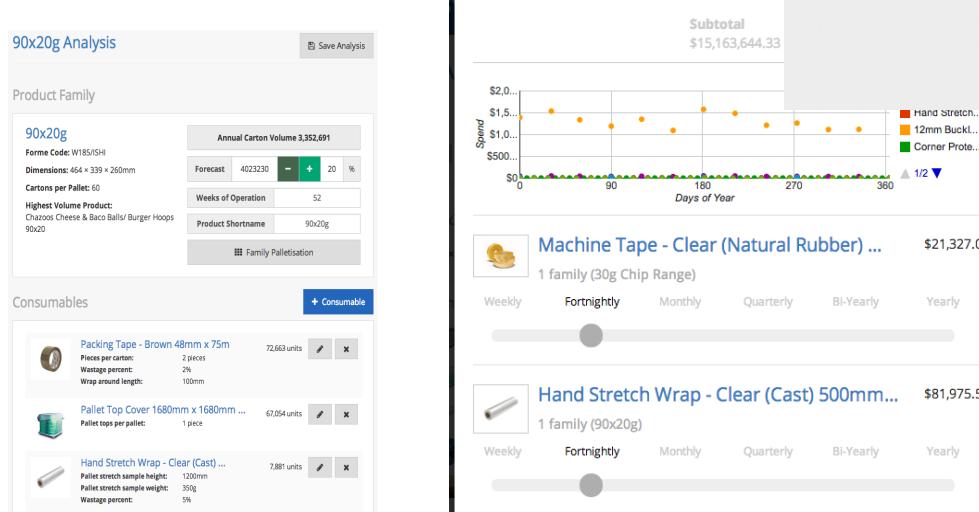
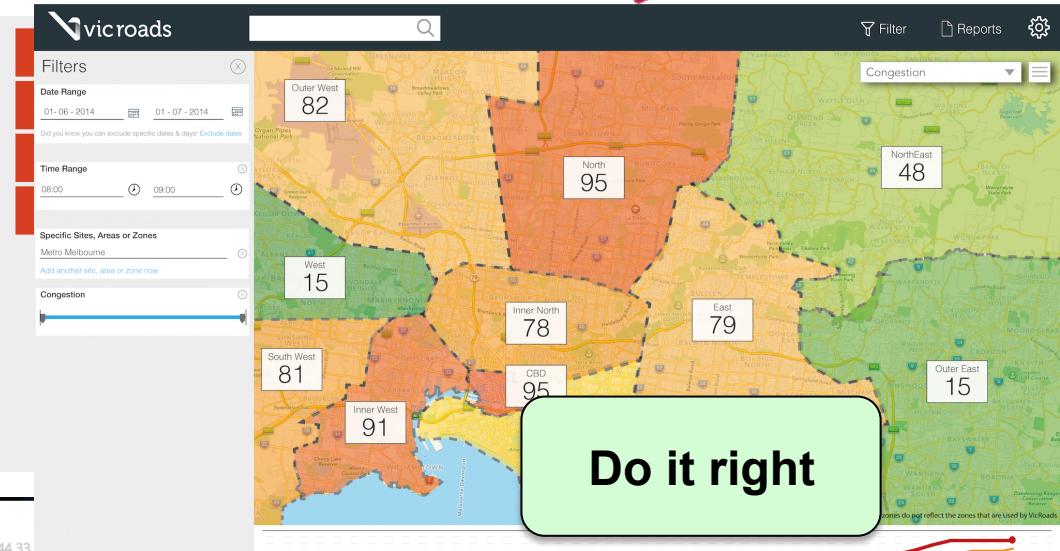
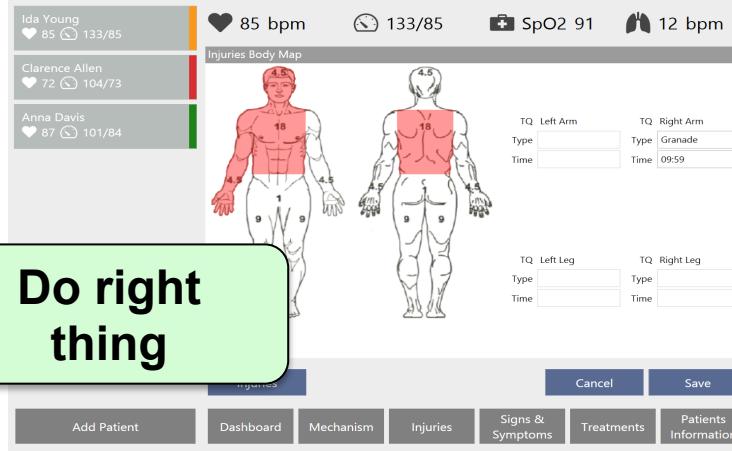
- Part of ARC Linkage w CA Labs
- Want to emulate large-scale deployment environment – possibly 000s systems e.g. for testing CA identify manager system, PeopleSoft ERM system, Twitter server, legacy mainframe clients, ...
- Record interaction “traces” e.g. LDAP binary protocol, PS SOAP messages, Twitter JSON, Mainframe text msgs
- When get request message, use clustering/distance functions/translation functions to synthesize response
- 3 patent applications to date



## Building Australia's SE research / education / practice

- Support high-quality SE research
- More Industry engaged learning @ SUT (and elsewhere)
  - Industry placements
  - Capstone projects
  - Industry-oriented exemplars in units
- NICTA/SUT Software Innovation Lab
  - Applied R&D with software companies
  - Joint investment in projects
  - Industry-based PhDs, UG students, researchers

## Software Innovation Lab - Various Projects



## Summary

- We need to (continue to) enhance SE research, education and practice in Australia
- We need to work together to do this
  - Education, research and practice
  - Some of our current approaches include
    - Practical-oriented SE research e.g. mobile app generation, software security engineering
    - Industry engaged learning; new educational assessment approaches incl. ePortfolios + Open Leaner models
    - The Swinburne / NICTA Software Innovation Lab e.g. various software R&D projects, practical processes, tools done joint with SE industry

Questions / Discussion

jgrundy@swin.edu.au



Swinburne  
►think forward