Legacy Software Systems
Should I stay… Or, should I go?

Dr Julian Bass, Computer Science and Software Engineering
Should I stay…?

• Legacy systems
  • Increasing cost of maintenance
    • 40 year lifetime cost £800M
  • Vulnerable to vendor market strategy change
    • End of life
  • Increasing cost of repair and enhancement

• Vendor lock-in
Should I stay… Or, should I go?

- Should I stay…
  - Increasing risk of catastrophic failure
    - Risk to beneficiaries and users
    - Reputational damage
  - High severity risk

- Or, should I go?
  - Forced by regulatory change
    - GDPR?
  - Impossible to adapt
  - Unthinkable throw away costs

New Regulations
Or, should I go?

- Migration Justified by
  - End-of-life
    - End of vendor support
  - Business case
    - Clear business benefit
  - Need for change
    - New features needed
  - Need for
    - multi-channel user support
    - micro-service architecture solutions
    - Containerised solutions
Salford Software Lab - Benefits

- Reduce migration project duration
  - Enhance productivity
- Improve speed of assessment
  - Faster decision-making
- Mitigate migration risk
- Second opinion on brute-force manual processes
- Low cost of entry
- Ease of exit, if not satisfied
Salford Software Lab - Process

• Proven Innovation Pipeline
• 5 Stage Innovation Process
  1. Study your challenge, in your context
  2. Proof of concept
  3. Minimum viable product
  4. Deploy and refine
  5. Productise and scale
• Or, choose your own approach
Salford Software Lab - Technologies?

• Technology options to meet your needs
  • Language (semantic) analysis
  • Source code visualisation
  • Cross compilers
  • Code translators
  • LLVM front-end technologies
  • SaaS and cloud architectures
    • Multi-tenancy
    • Micro-service solutions
    • Elasticity