

# Outline

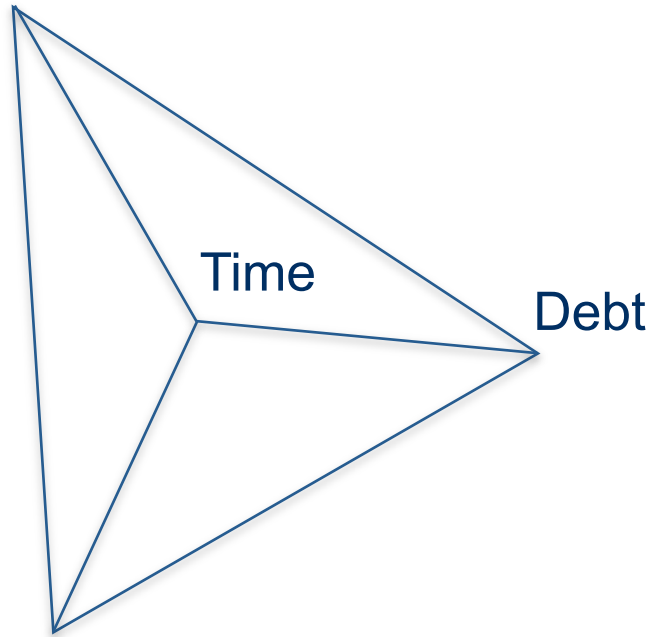
---

- Introduction
- Concepts
- Measures
- Management
- Conclusions

# Introduction

---

Cost



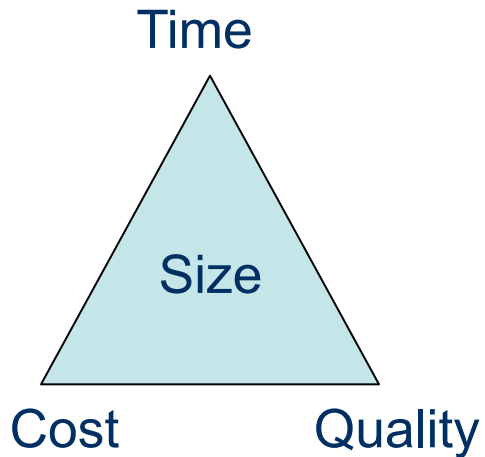
How can businesses and IT organisations manage the long term sustainability of applications while constantly updating them?

By being aware that the business decisions they make in relation to their applications space will incur business costs that can be identified measured and managed.

Quality

# Basic Development Dimensions

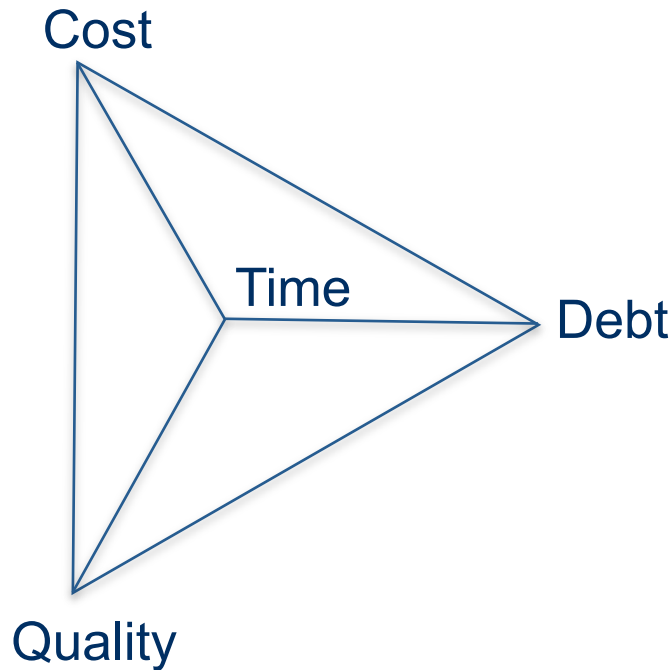
---



- Best estimates are created using control databases and models to determine the optimum cost of development based on time available and quality required within a certain period after delivery.
- Change must be accommodated, but often with no change to budget or timescale and often at short notice.
- Pushing the envelope has the effect of introducing errors and deviations from standards
- The current release may be acceptable but what troubles are being built-up?

# The fourth dimension

---



- Adding considerations of technical debt to time cost and quality gives us the tetrahedron
- This can be assessed at every level from sprint to iteration to waterfall enhancement and finally to application support costs
- Tetrahedrons stack up to the end to end Total Cost of Ownership

# Managing Technical Debt

---

"As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it."

— Meir Manny Lehman, 1980

- The idea of technical debt is far from new, but in the rush to keep costs down during development it is too often simply ignored
- It has also been said that failure to address increased complexity results in the effective payment of interest on the debt
- The challenge is to get business to recognise the problem
- Work with the business to quantify it
- Work as a team to manage it
- Short term tactical cost cutting can often result in long term strategic costs

# Measures

---

- One size does not fit all
- Measurement must make sense to the business
- Typical measures include
  - Mean time to defect
  - Percentage down time
  - Total number of defects cleared in a year
  - Total cost of defect clearance
  - Code complexity measures
- The important part is to put a value on it
- Once the business sees a value, that can be managed using quantitative means



# Management – Organisational

---

- Acknowledge the problem and sponsor debt measurement and management
- Define metrics which link technical debt to cost of the debt
- Determine the total debt (see *Applications* below)
- Determine how much debt you want to live with
- Set objectives for management and review them annually
- Review the governance process
- Consider a link to the IT-CMF
- Consider using code parsing software to identify issues
- Change the behaviours in the organisation towards prevention rather than cure
- Make career progression for Commissioning Managers and Project Managers dependent in part on debt control
- Initiate projects to reduce the debt
- Review and change metrics if appropriate

# Management – Applications Support

---

- Identify the debt within an application
- Use code parsing tools, if available, to identify issues for clearance.
  - If tools aren't available do a cost benefit analysis and consider purchase
- Quantify the cost of non-clearance
- Quantify the cost of clearance
- Specify the acceptable level of added debt per release
- Track debt levels and compare with objectives
- Specify a point where the debt has to be reduced
- Implement a project to manage the debt.
- Review and change metrics if appropriate



# Management – Applications Development

---

- Amend the estimating process to ensure that clearance of debt is planned for at the start of the project
- Ensure technical reviews during development trap and clear architectural non-conformances
- Use code parsing tools during the project to ensure that non-conformances are identified and cleared before the software goes live.
- Hold commissioning managers and PMs responsible for levels of technical debt
- In Agile manage the technical debt as a series of stories and make organisational decisions about when and if to address it.
- Hold post implementation technical reviews with implementation dependent on passing the technical review gate.
- Review and change metrics if appropriate

# Conclusions

---

- There are compelling pressures on our industry to reduce costs
  - Increase productivity
  - Deliver change on a vast scale
  - Cope with endlessly changing applications and keep them running for many years
- The concepts in this paper are a plea to everyone to be honest about the costs of all of the above and to attempt to measure the ongoing cost of technical debt and to take steps to manage it.
- This requires changes in mindset at all stages in an organisation but quantifying and grasping this bullet is the way to build long term value for corporations using to-days complex applications.

# David Consulting Group Ltd

---

For more information visit our website, e-mail or call.

[www.davidconsultinggroup.com](http://www.davidconsultinggroup.com)

## Contact:

Alan Cameron

- [Acameron@davidconsultinggroup.com](mailto:Acameron@davidconsultinggroup.com)
- +44 1253 290370 Office
- +44 7825 271 344 Mobile