

# CA PLEX UPGRADE STUDY

6A

Matthew Morris

Desynit



**ADC AUSTIN**

**MKS**

**WEBSYDIAN™**

**cm** FIRST



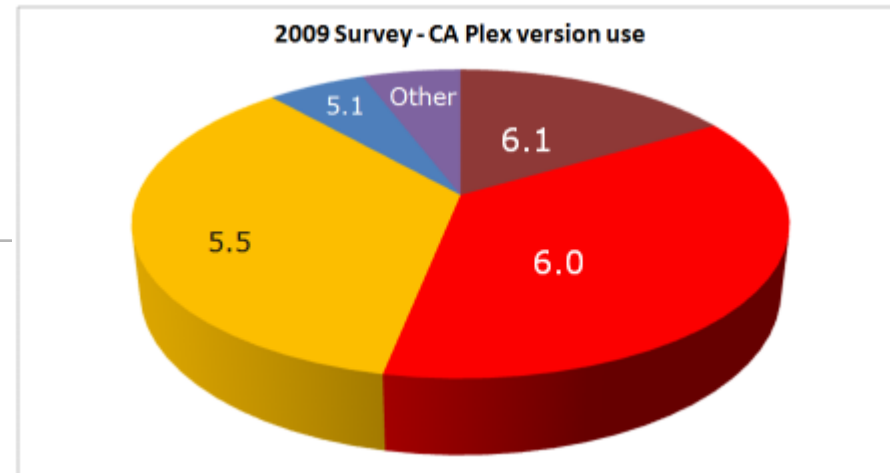
# Desynit

---

- > Founded in 2001
- > Based in Bristol, U.K
- > Customers worldwide
- > Technology Mix
  - 2E/Plex
  - Java & .Net
  - Web & mobile applications
- > Contact
  - [Matthew.morris@desynit.com](mailto:Matthew.morris@desynit.com)

# So why isn't everyone on the latest release?

- > Current Plex systems are not being adequately maintained
  - 47 % use unsupported version
- > Majority of effort is focused on tactical issues
- > How can we break out of this cycle?
- > Upgrade your software development lifecycle:
  - Respond to business need
  - Be confident you are controlling the risk of change
  - Focus effort on making, rather than implementing, changes



# Prepare for change

---

## > Baseline environment

- Test vs. New build

## > Locate differences

- Generated code (action diagram and source)
- Built objects (# of objects and DB schema)

## > Use/Build/Update Test Packs

- Adopt automated testing

## > Manage all assets

- Place generated code under source control
- Use SCM/ALM tools to manage objects (and Plex models)

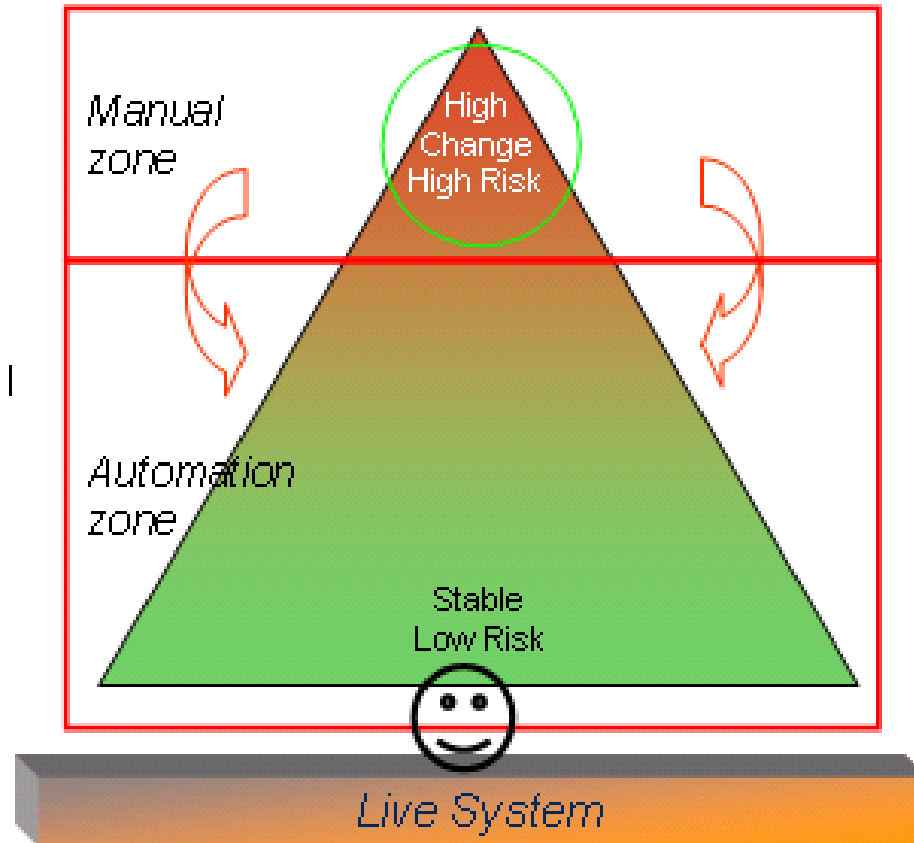
Impact Analysis  
**DEMO**

# Build & Qualify Application

---

- > Impact of code change is understood, still need to test!
  - CA runtime changes
  - Microsoft & 3<sup>rd</sup> Party runtime changes
- > Build the objects and move them to the test environment
  - Make sure procedures are followed EVERY time
  - Secure copies of the environments for future use
- > Reduce the time spent on these tasks
  - Automated testing tools
  - Change management tools

# Our application is stable: Perfect for automated testing



# What have we upgraded?

---

## > Our control of the system:

- All assets managed
- Documentation via tests

## > Our ability to:

- Respond to business need
- Be confident you are controlling the risk of change
- Focus effort on making, rather than implementing, changes

# Get back in control

---

## > Anxious about upgrading?

- Put effort into your supporting processes

## > The support needed for an upgrade is the same as for any change.

- You can't ignore the risks

## > Each upgrade has unique circumstances

- The approach is the same; compare, test, deploy

## > Help and advice?

- [matthew.morris@desynit.com](mailto:matthew.morris@desynit.com)