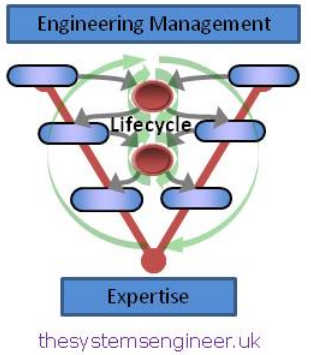
30th Annual INCOSE international symposium

Virtual Event
July 20 - 22, 2020



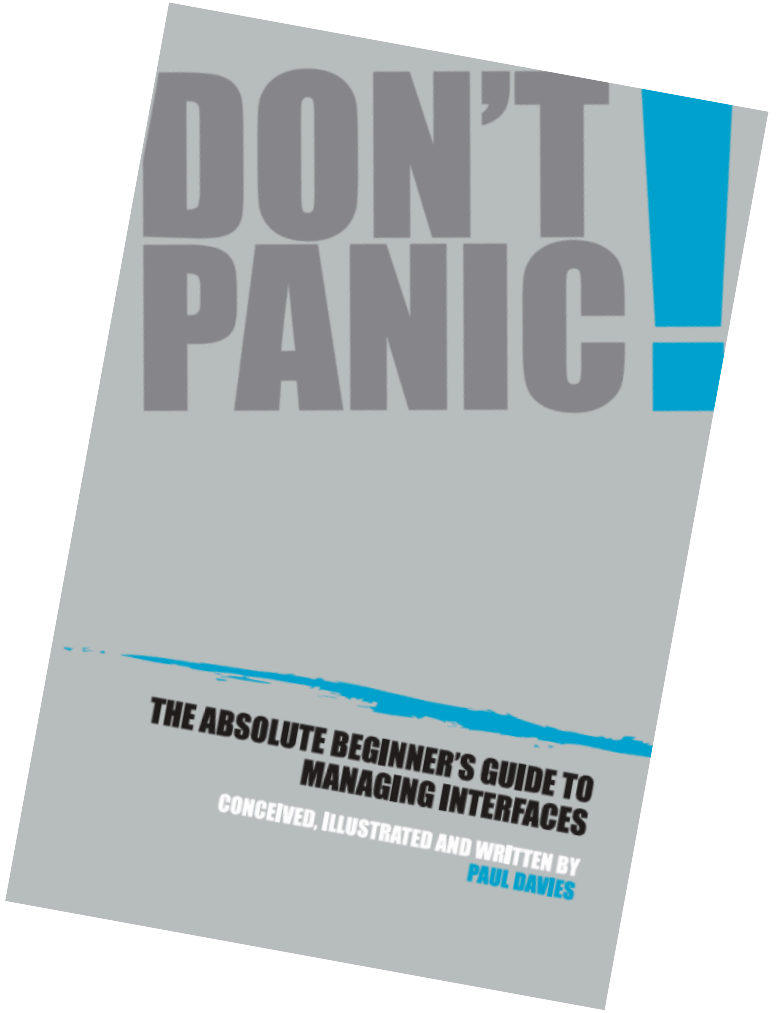
Interface Management – The Neglected Orphan of Systems Engineering

Paul Davies
paul@thesystemsengineer.uk

www.incose.org/symp2020

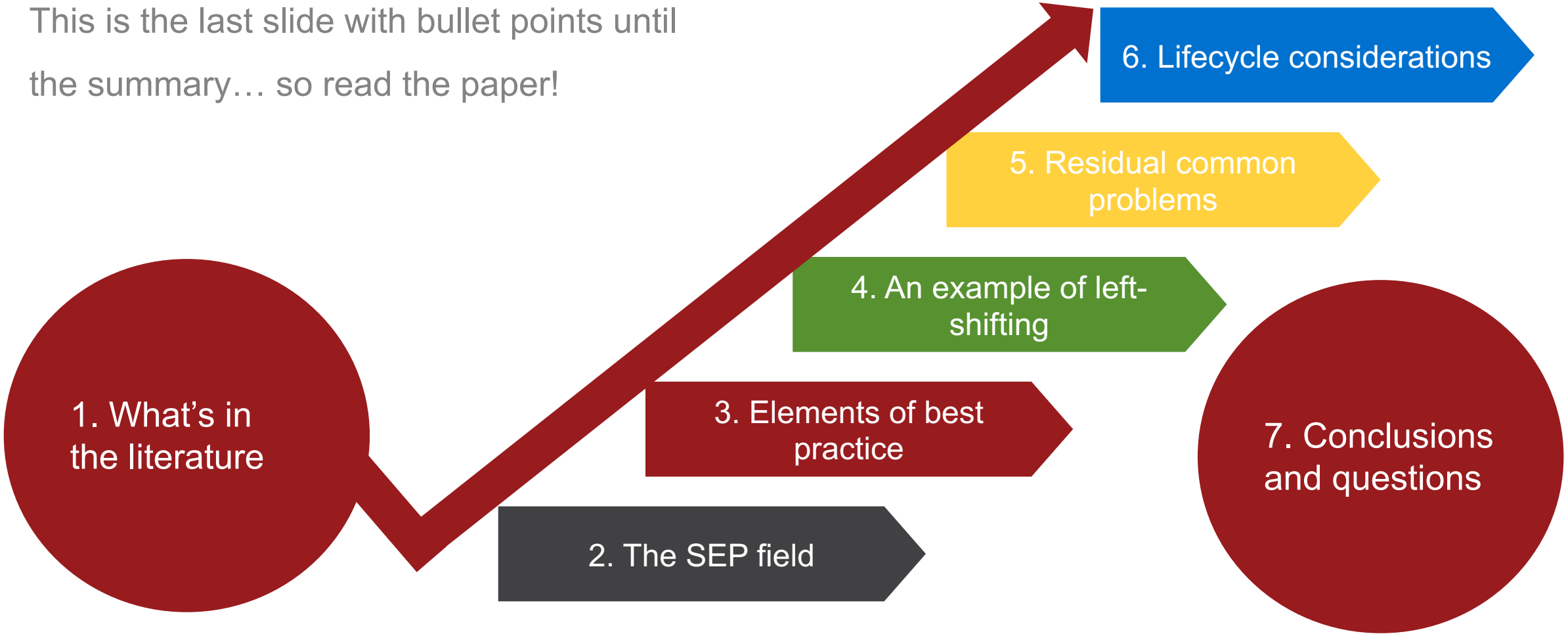
Aims of the paper

- To challenge the perception of an engineer as someone who ignores the world outside their System Element
- To remove the excuse “There’s no training or guidance on interfaces”
- To promote a whole-lifecycle view of interfaces, and left-shift their consideration in architecting

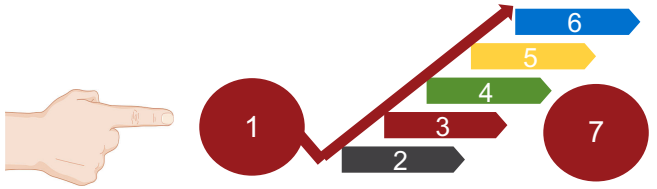


Outline

This is the last slide with bullet points until the summary… so read the paper!



What's in the Literature



IEEE Standards

IEEE Std 1220™-2005
(Revision of
IEEE Std 1220-1998)

1220™

IEEE Standard for Application and Management of the Systems Engineering Process

IEEE Computer Society

Sponsored by the
Software and Systems Engineering Standards Committee

IEEE

3 Park Avenue, New York, NY 10016-5997, USA

9 September 2005
Print: SH95334
PDF: SS95334

ANSI/EIA Standard

Processes for Engineering a System

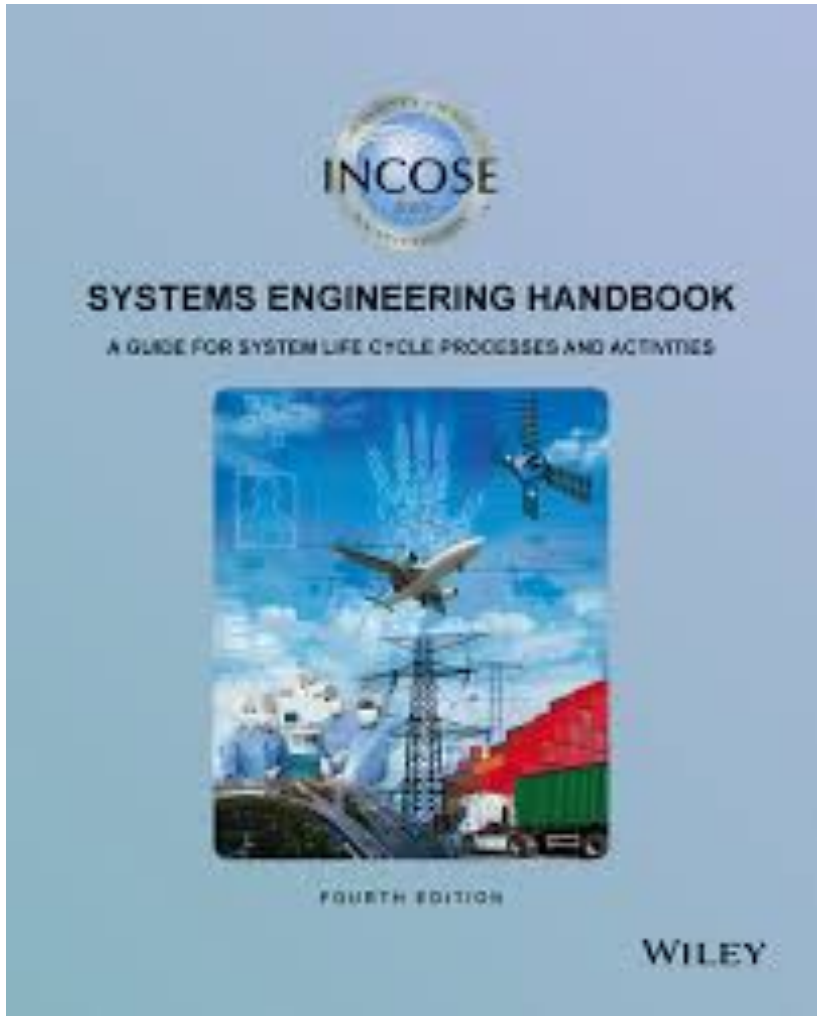
ANSI/EIA 632

NASA Reference Publication 1370

1997

Training Manual for Elements of Interface Definition and Control

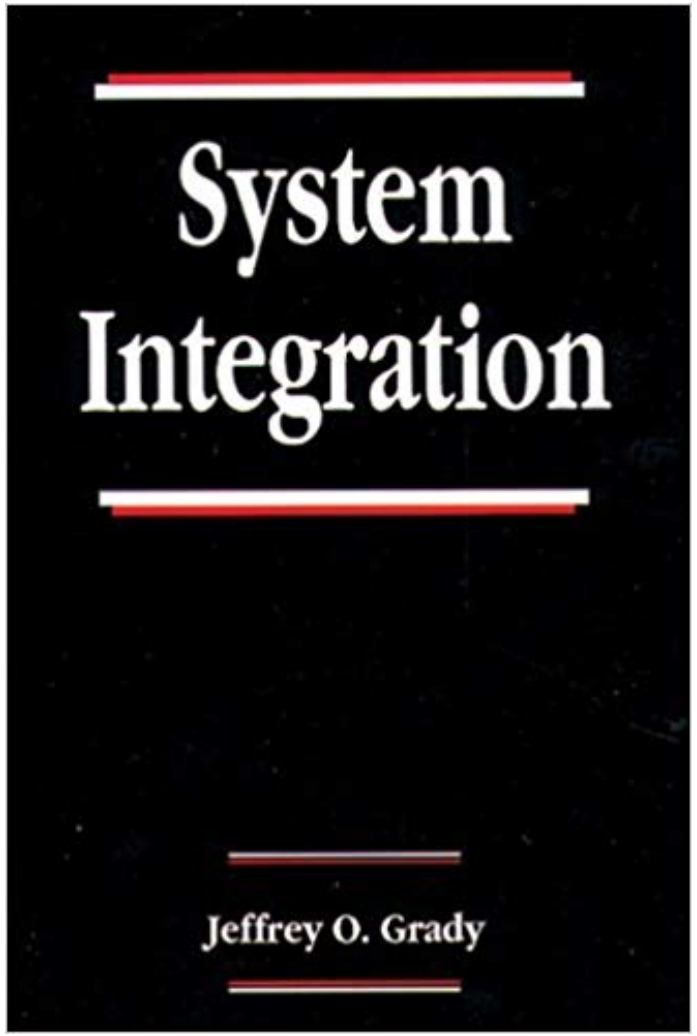
What's in the Literature

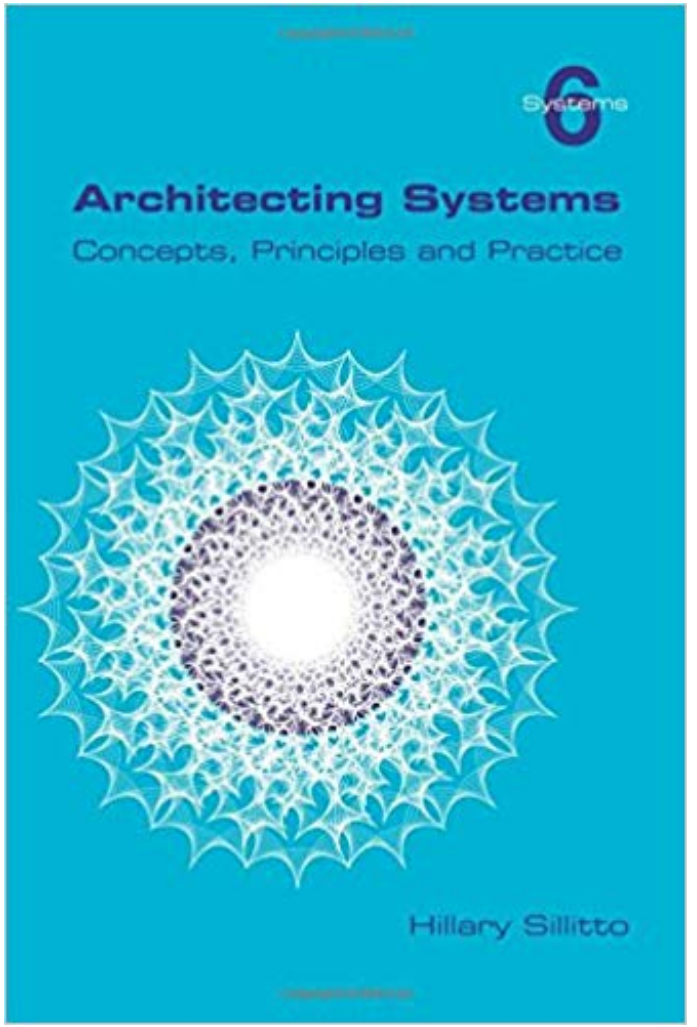


BKCASE SEBoK

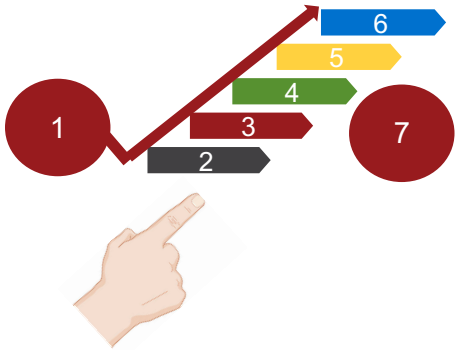
Guide to the
Systems Engineering Body of
Knowledge (SEBoK) version 1.9.1

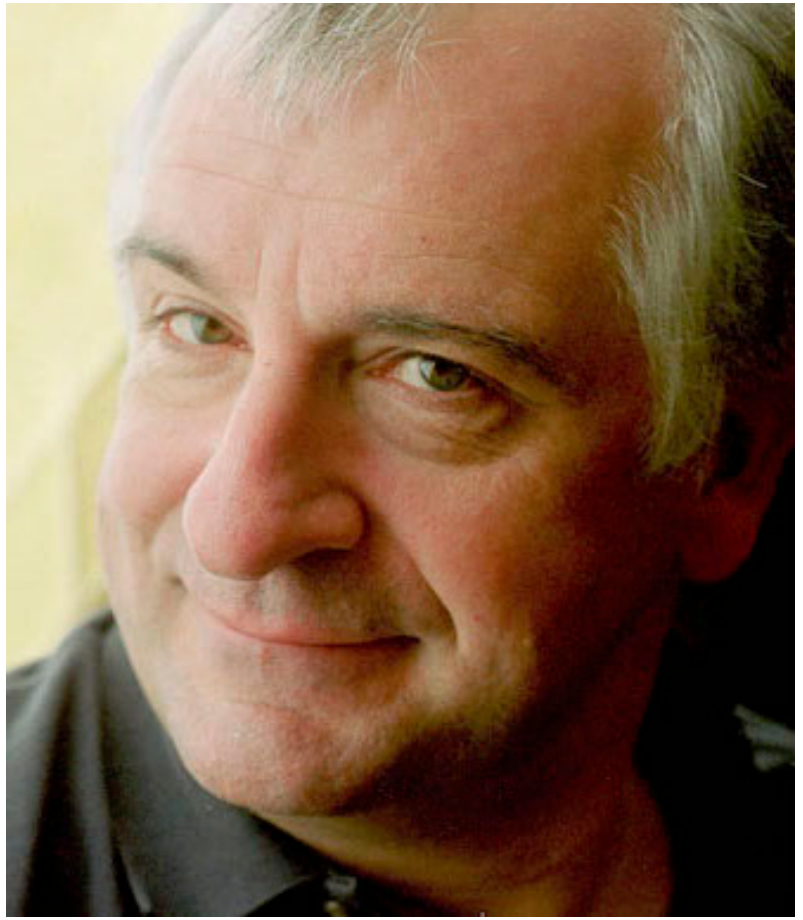
What's in the Literature



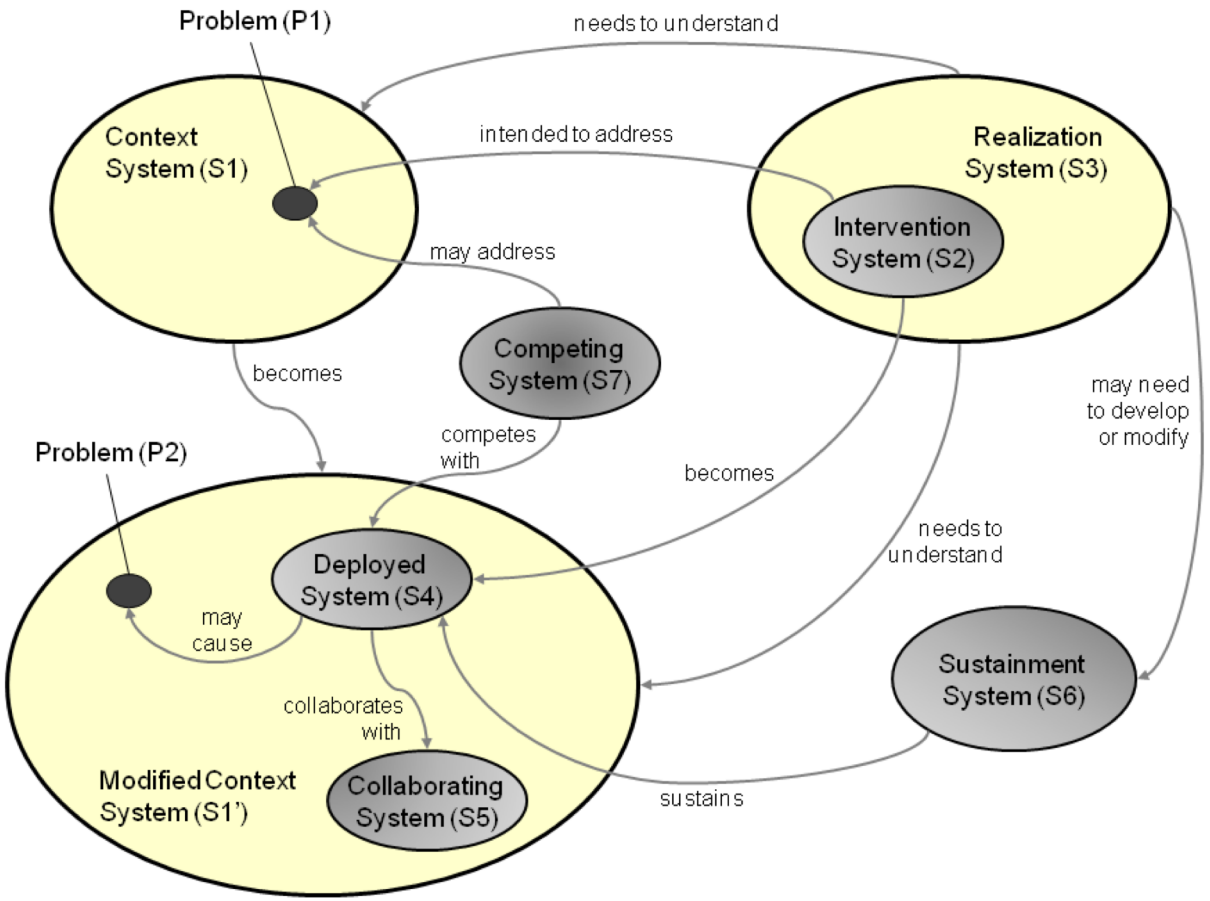


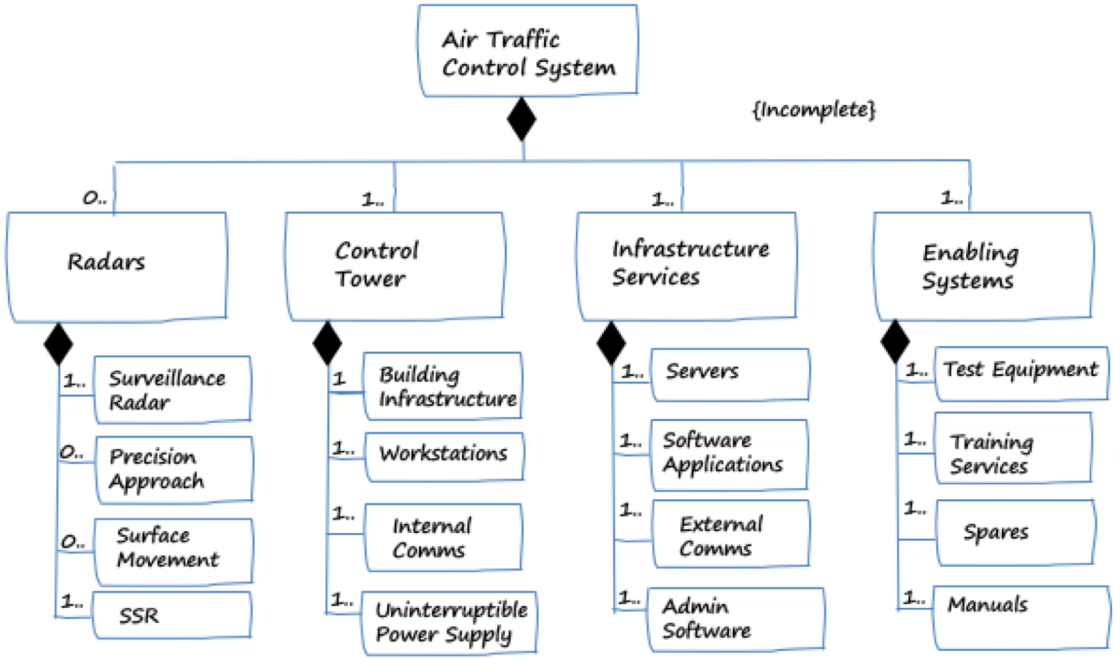
The “Somebody Else’s Problem” field



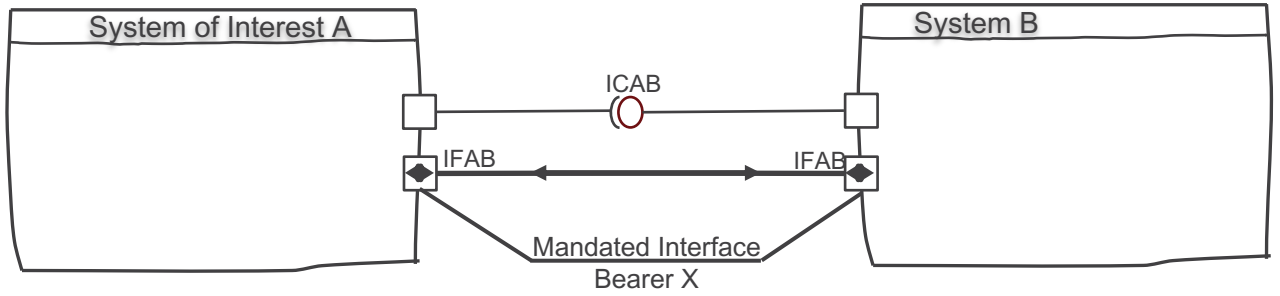


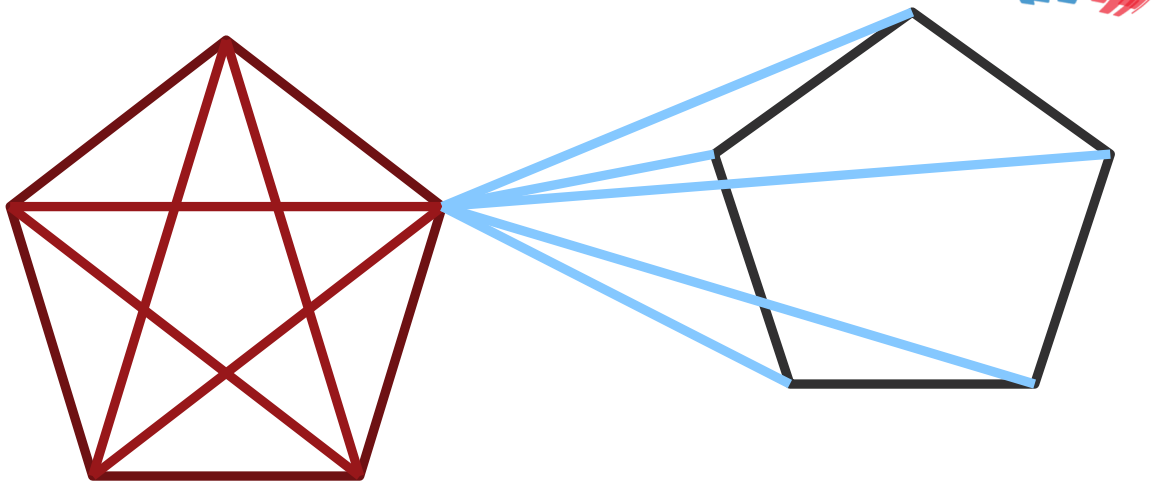
7 Samurai battle the SBS



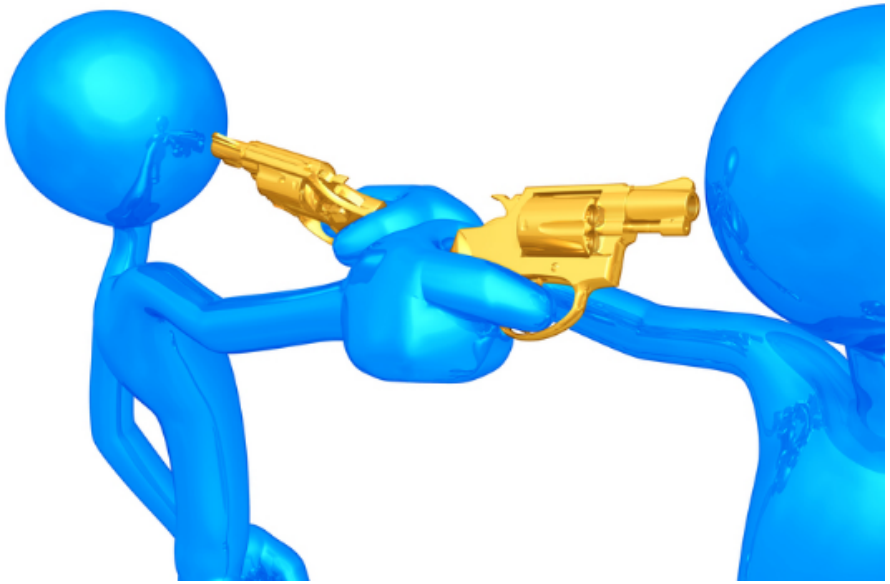


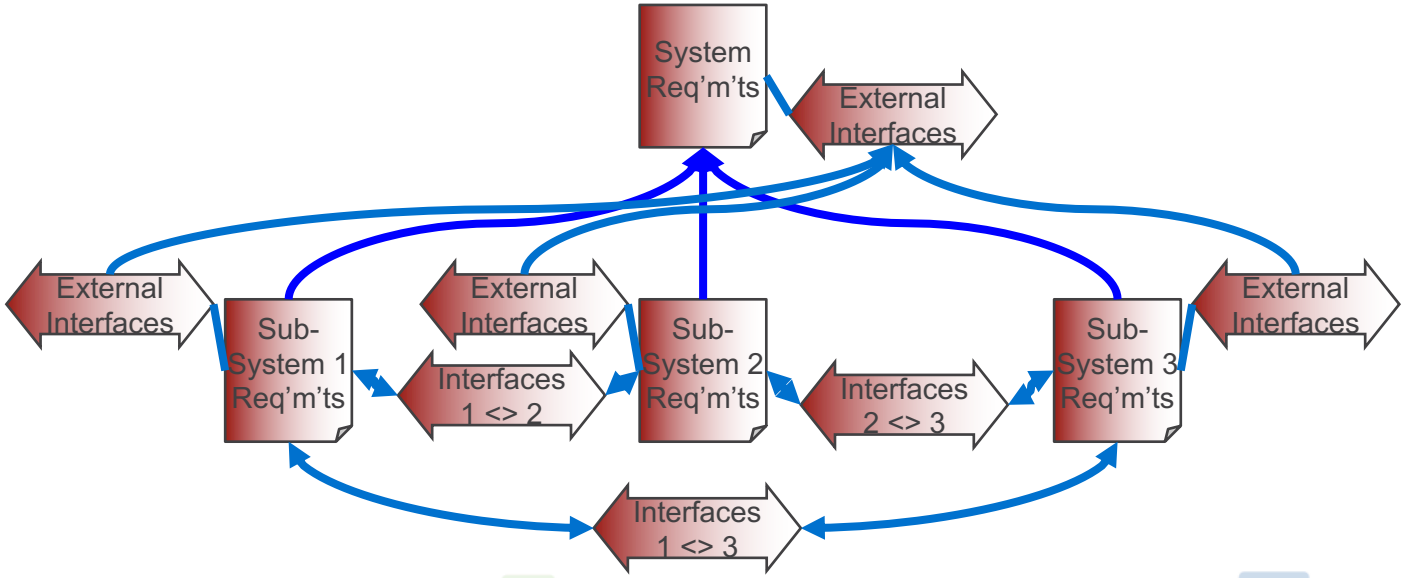
Why does it matter?



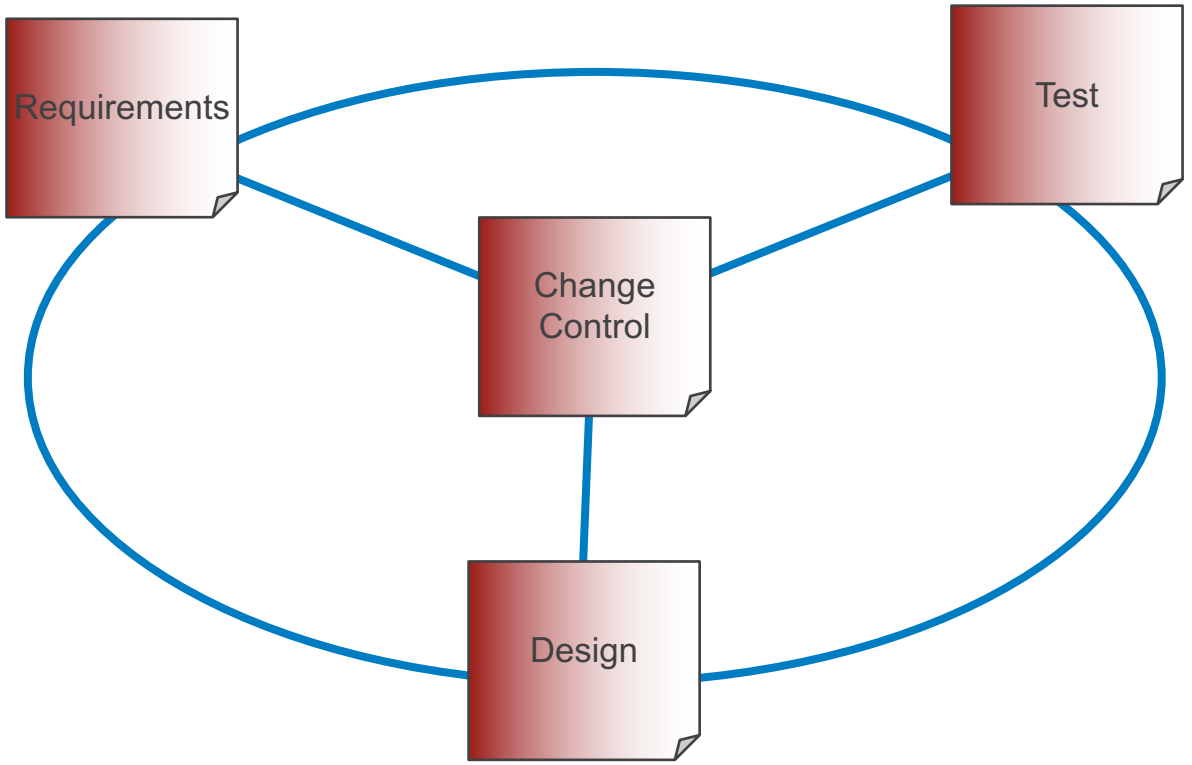


"System A shall interface to System B via bearer X"





Why does it matter?

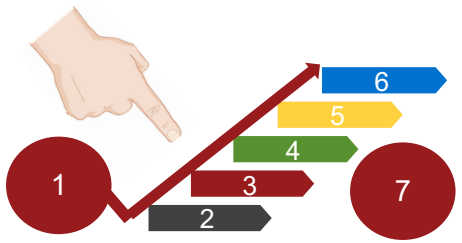


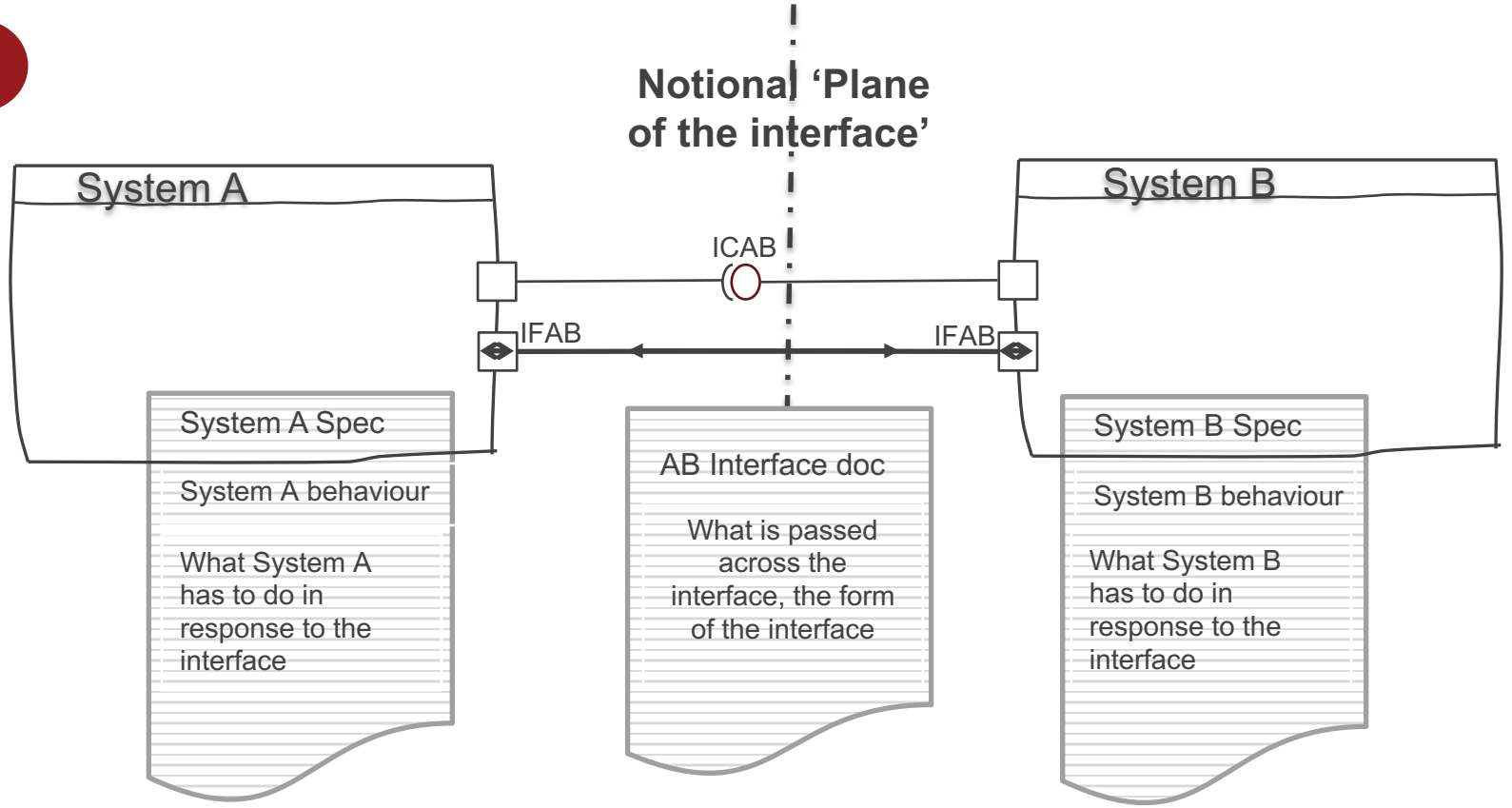
It’s not just software

Electrical voltage + current (+ spikes)
Vertical forces (time-varying)
Longitudinal forces due to friction
Heat
Flash arcing
Electromagnetic field flux (+RFI)
Vibrational forces (resonance?)
Shock (at joints)

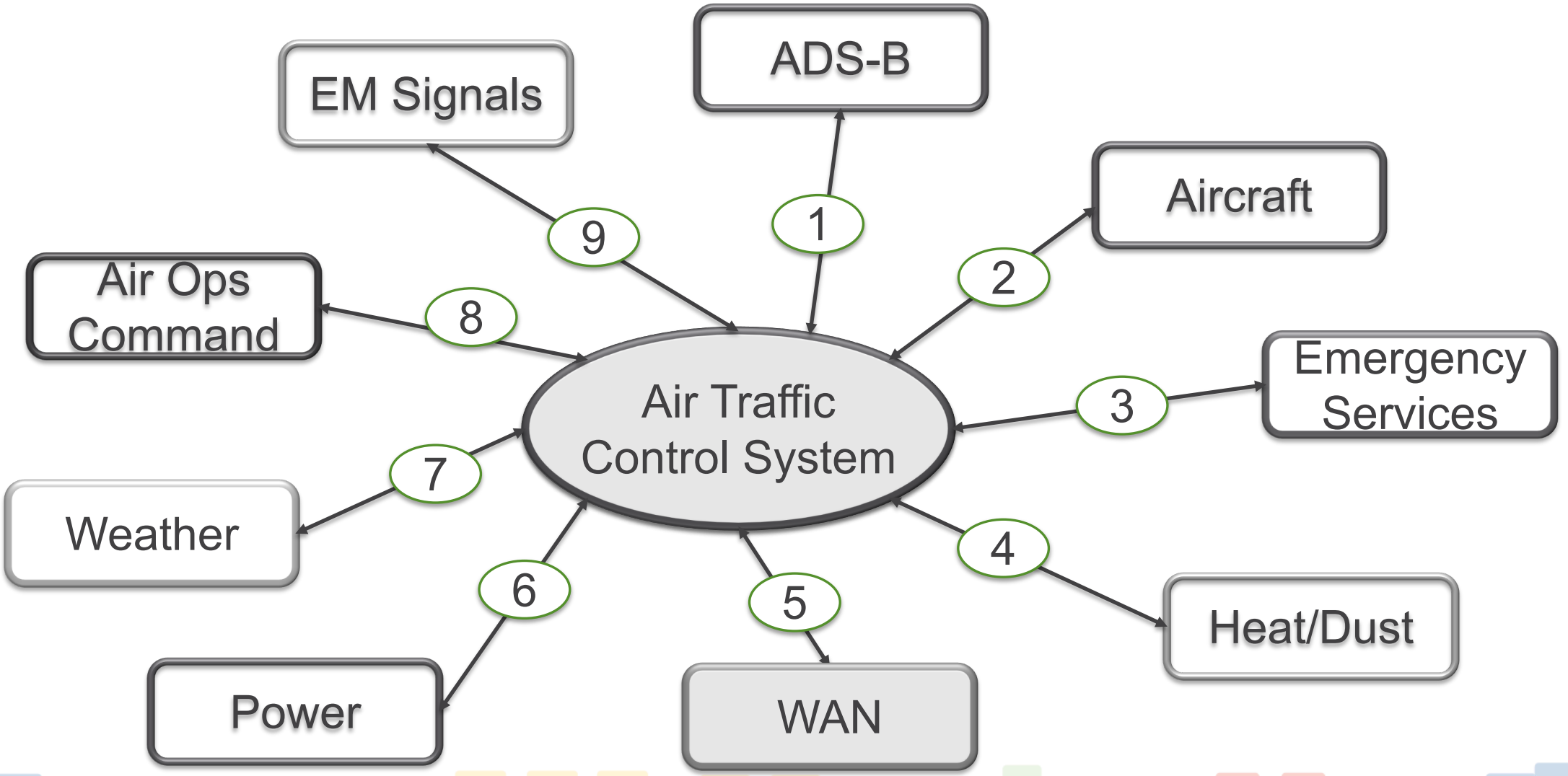
Moisture & salt deposition
Carbon deposits, rust, crud

Best Practice 1: the Separation Principle

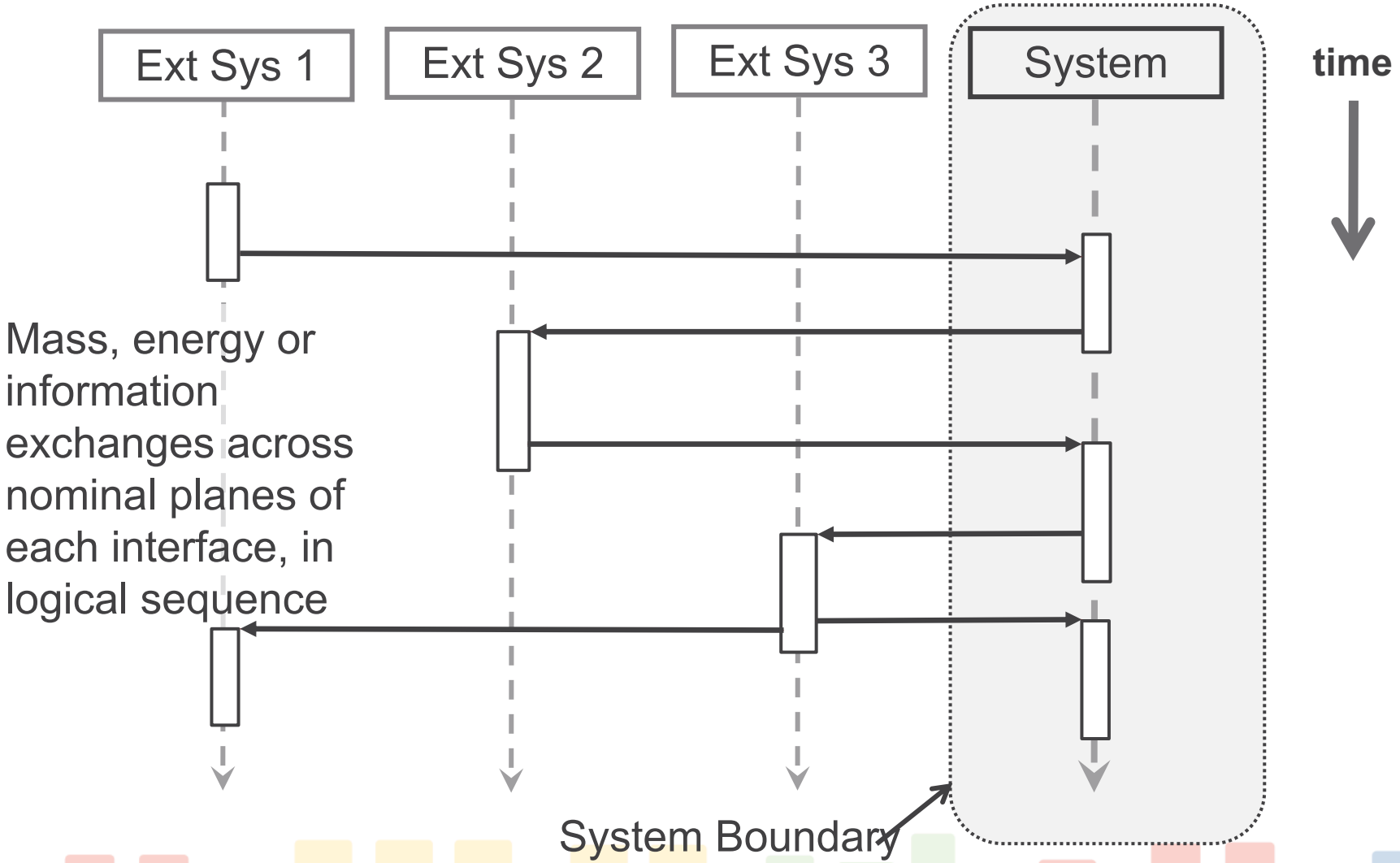




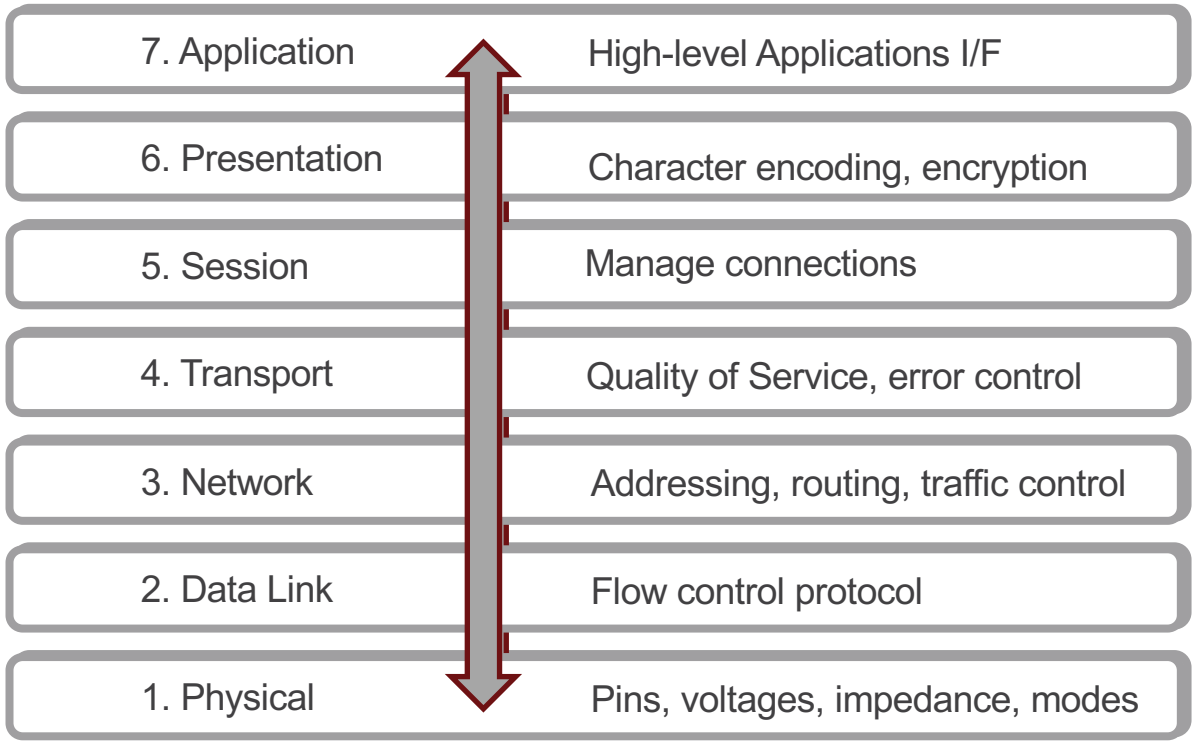
Best Practice 2: the Context Diagram



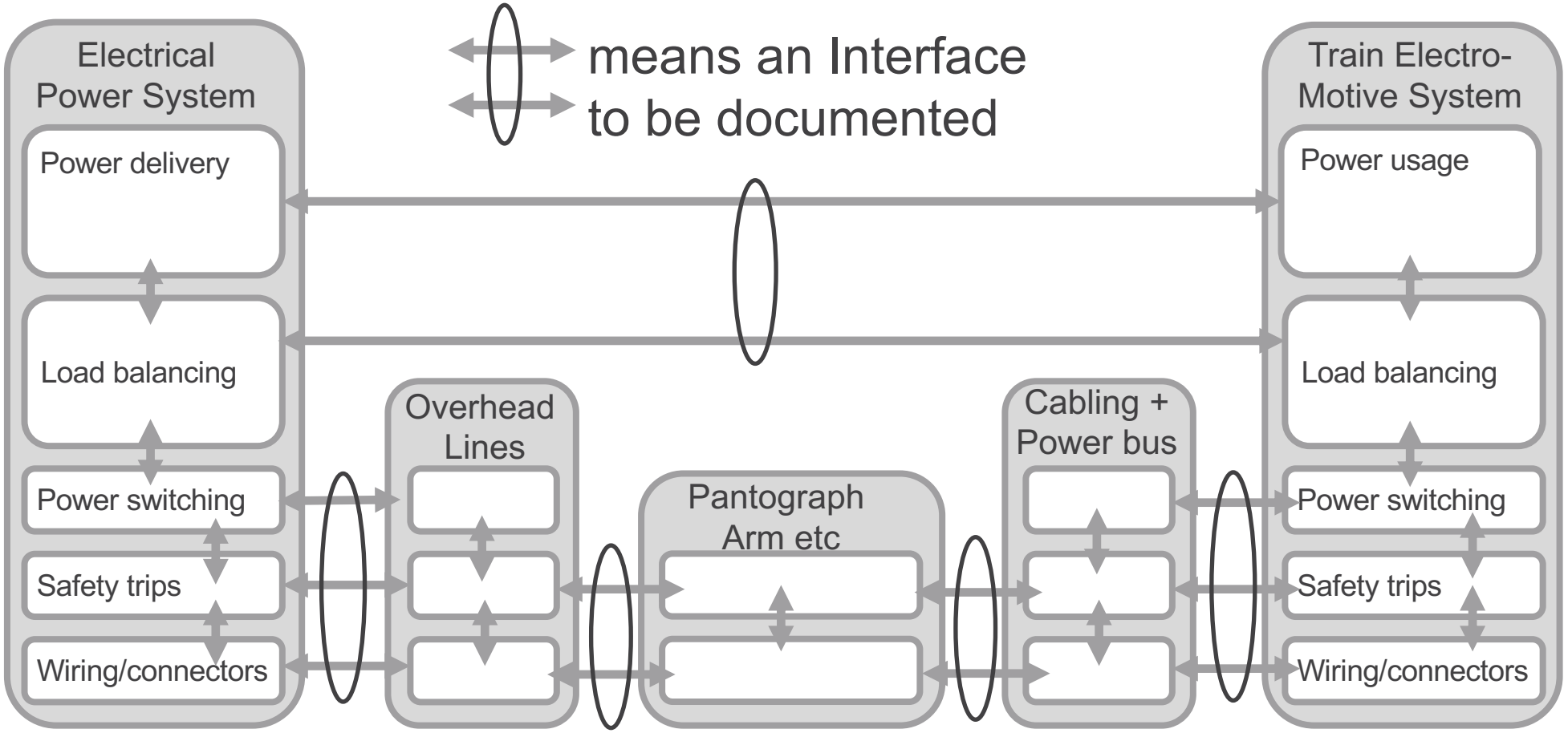
Best Practice 3: the Sequence Diagram



Best Practice 4: layered models as patterns

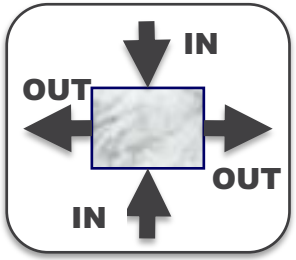


Best Practice 4: layered models as patterns



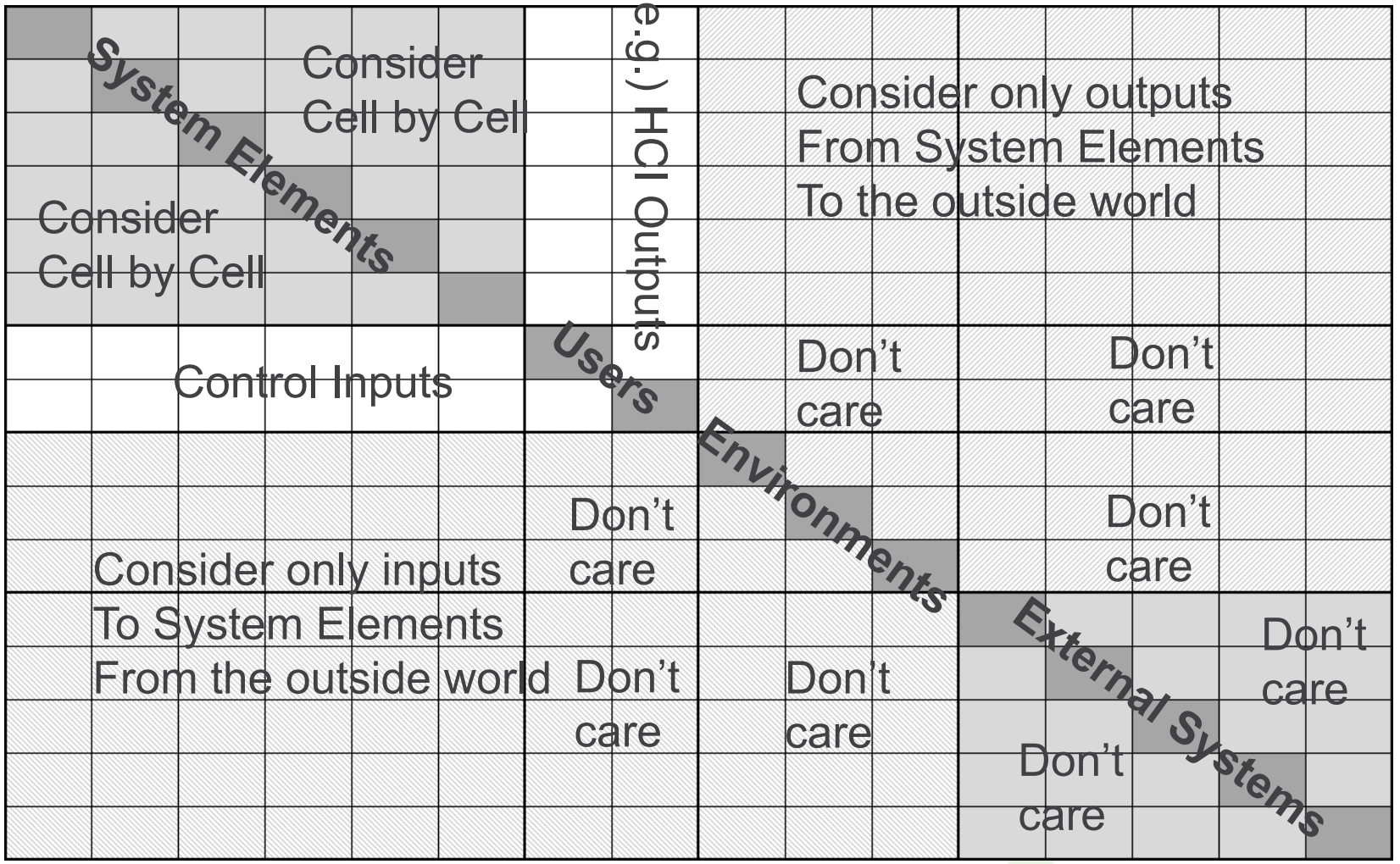
Best practice 5: black box N^2 chart

A -System Of Interest	B requires of A A provides to B	C requires of A A provides to C	D requires of A A provides to D	E requires of A A provides to E
A requires of B B provides to A	B - Users	C requires of B B provides to C	D requires of B B provides to D	E requires of B B provides to E
A requires of C C provides to A	B requires of C C provides to B	C - Physical Environment	D requires of C C provides to D	E requires of C C provides to E
A requires of D D provides to A	B requires of D D provides to B	C requires of D D provides to C	D - External System 1	E requires of D D provides to E
A requires of E E provides to A	B requires of E E provides to B	C requires of E E provides to C	D requires of E E provides to D	E External System 2

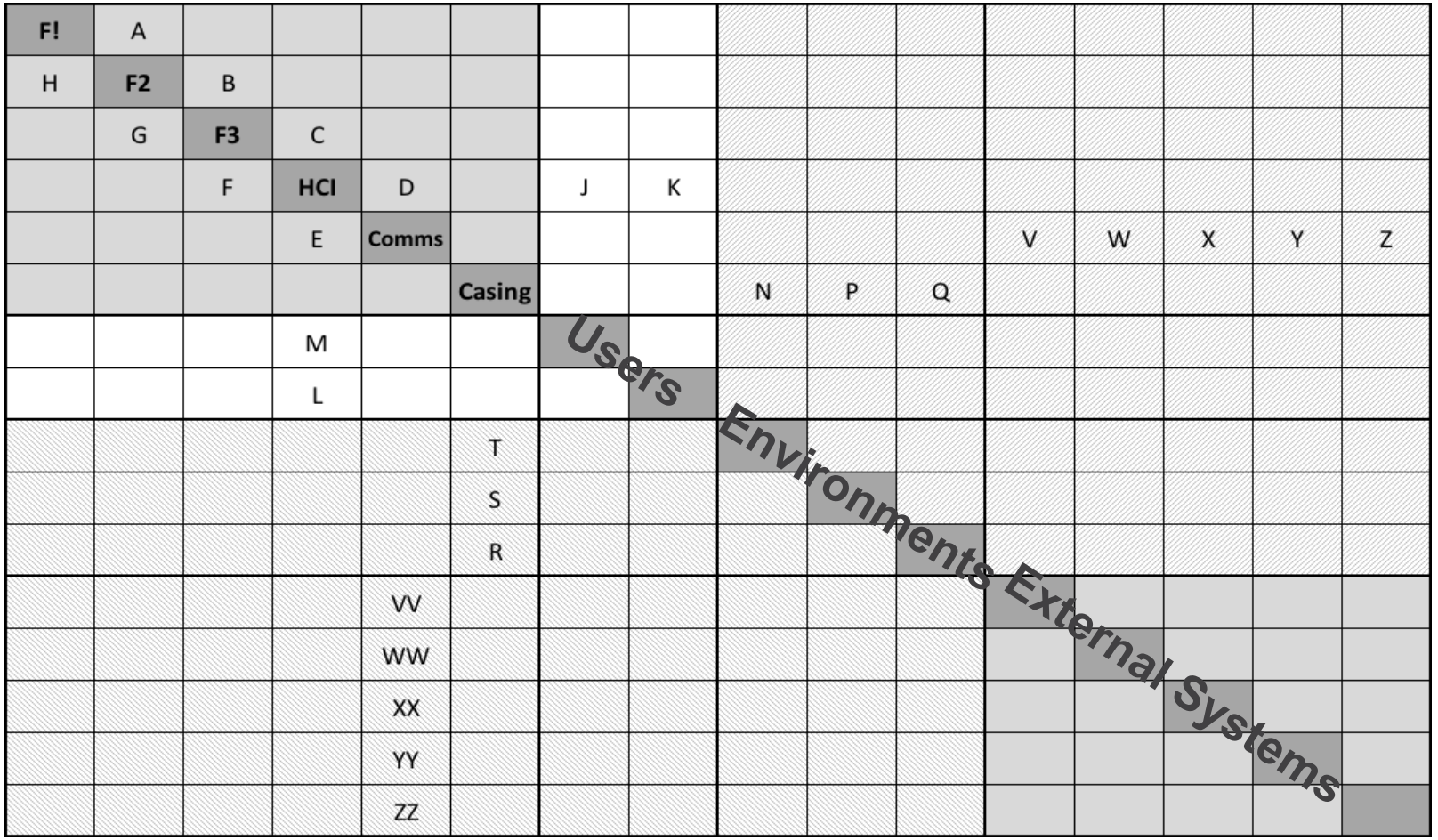


…etc

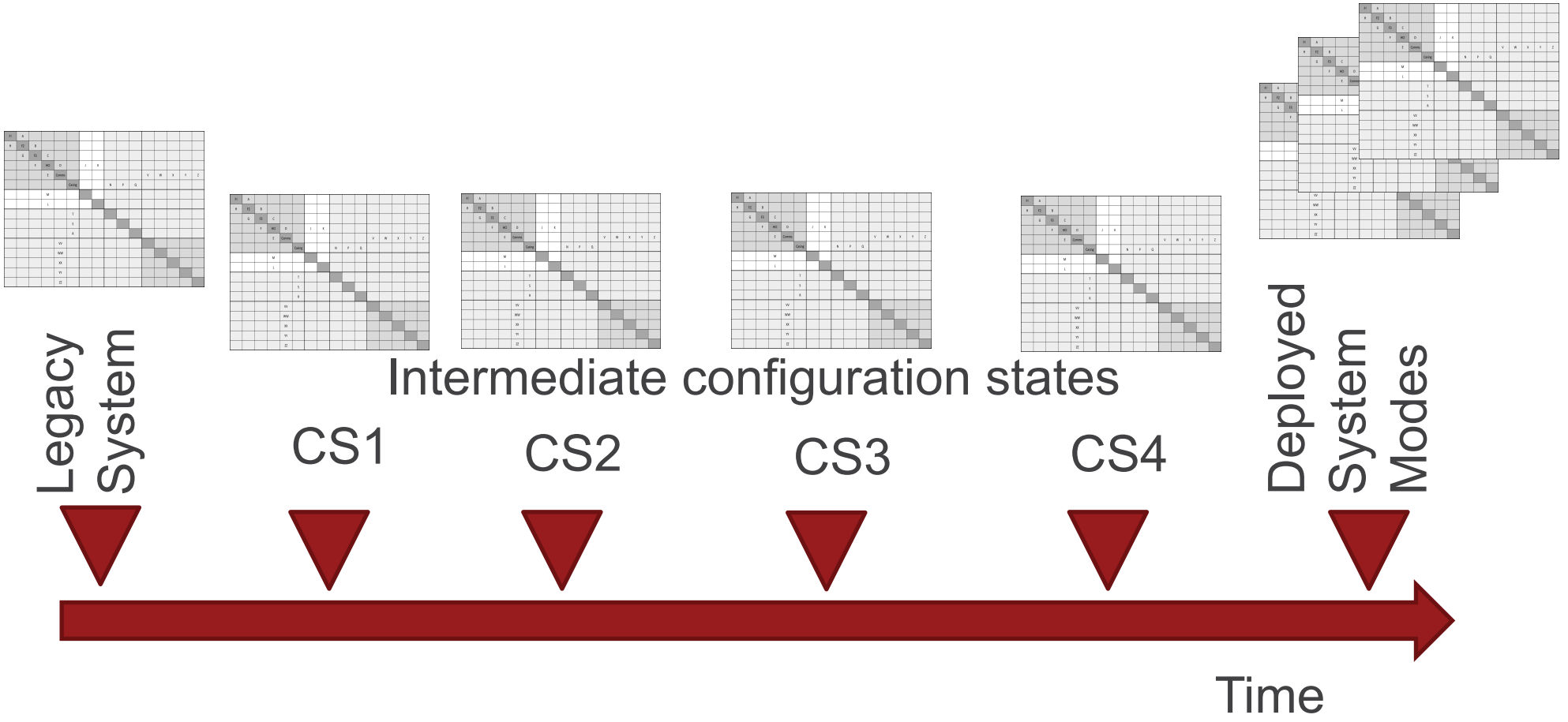
Best practice 6: white box N^2 chart



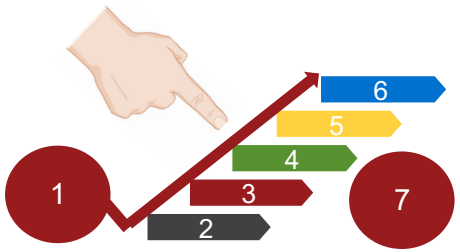
Best practice 7: optimised N^2 chart



Best practice 8: phased implementation N^2



Left-shifting…



Requirements → Architecting → Logical / Functional Design → Physical Design →

Becomes…

{Interface design at successive levels of analysis}

Requirements → Architecting → Logical / Functional Design → Physical Design →

Interface constraints

{Interface analysis feedback loops}

Pantograph example again

Electrical voltage + current (+ spikes)

Vertical forces (time-varying)

Longitudinal forces due to friction

Heat

Flash arcing

Electromagnetic field flux (+RFI)

Vibrational forces (resonance?)

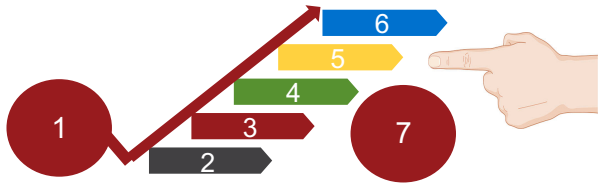
Shock (at joints)

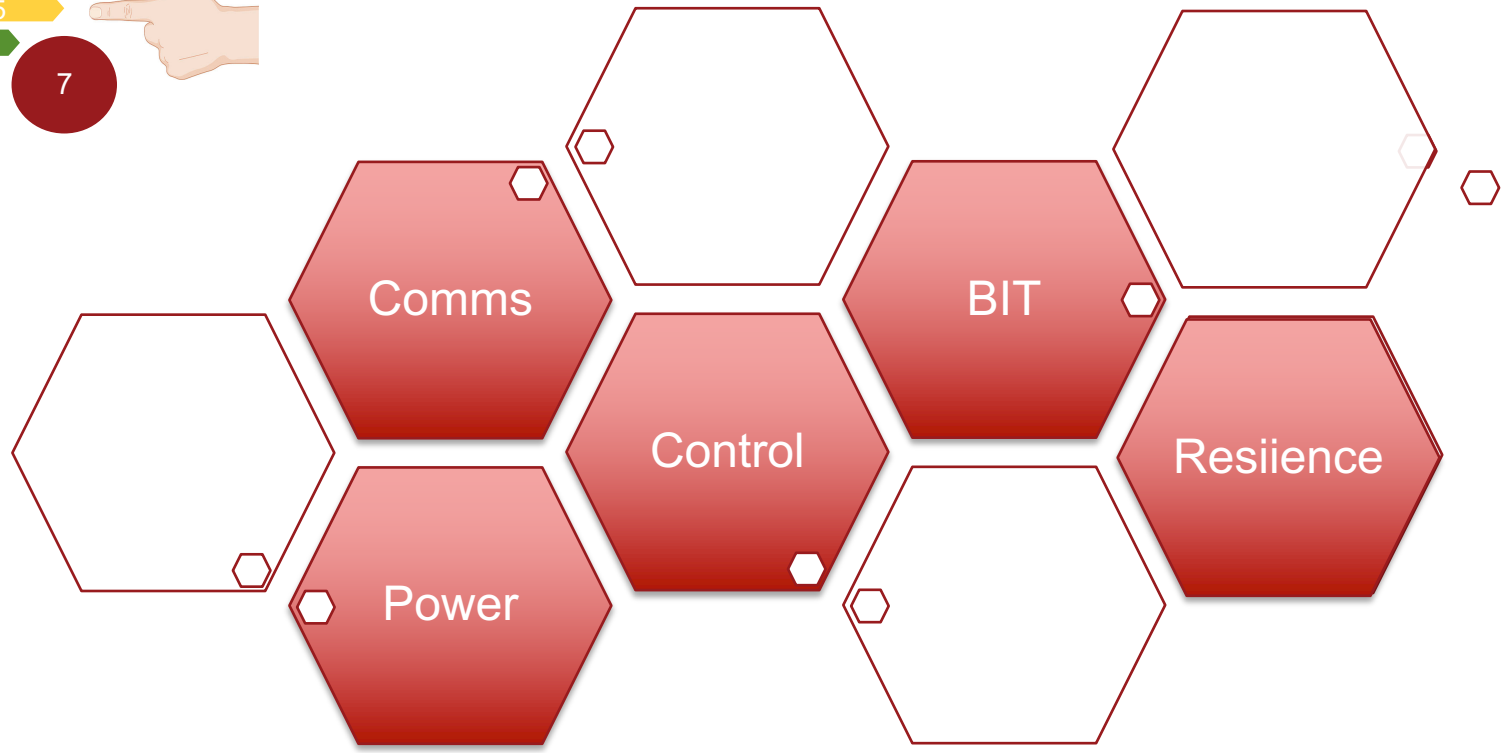
Moisture & salt deposition

Carbon deposits, rust, crud

The flows across the interface drive extra functional and non-functional requirements on the System Elements at each end

Residual architecting decision patterns for interfaces





Lifecycle of interface-based architecting

Draw the context

Identify the external I/Fs

Current vs Future

Agree way forward

OK?

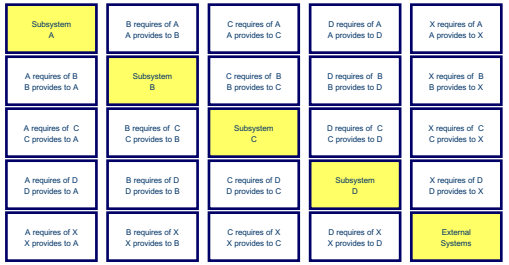
N

Y

Evaluate effect on interface decompositions

- Organisations
- Complexity
- Number
- Risk

Trial decomposition into subsystems

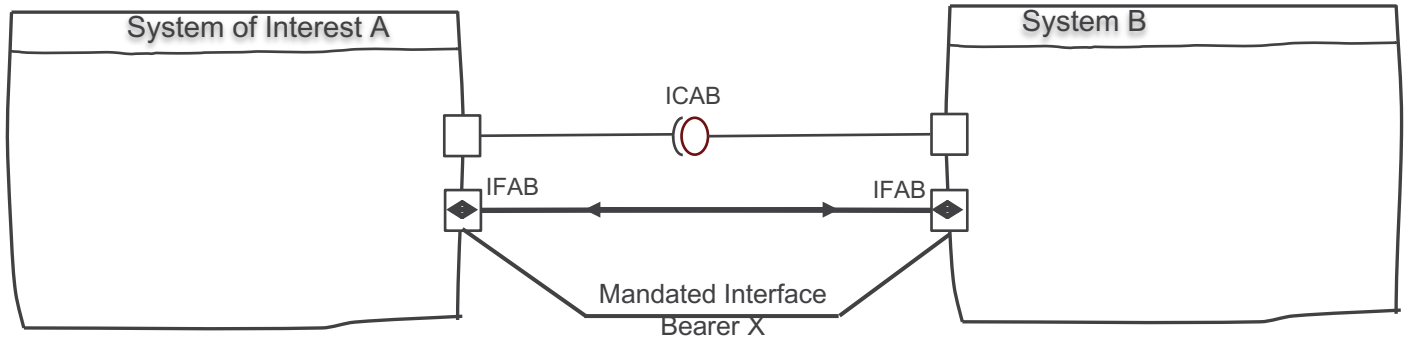


Generate Specification tree

Get on with it!

Phased testing

The requirement from hell, and future-proofing

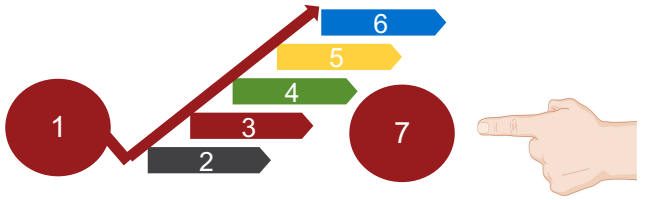


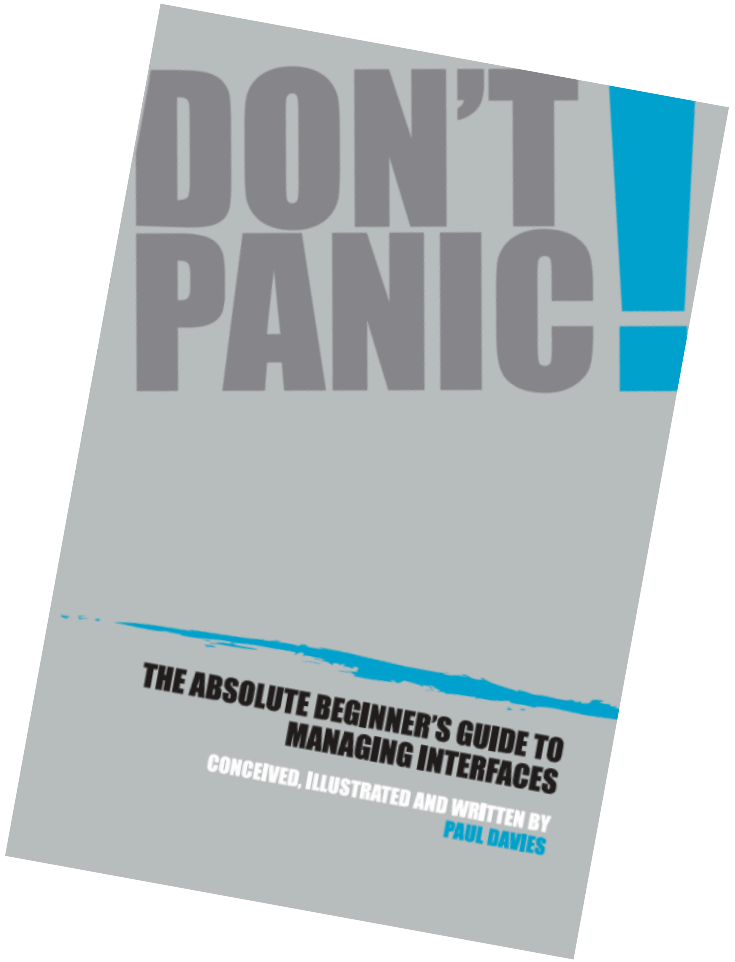
“System A shall interface to System B via bearer X”

- Bearer X in service; System B to be accepted
- System A envisaged, with interfaces
- IDD published for X; IRS created for A<->B
- System B accepted; ITT for System A uses IRS + IDD
- System A procured at lower cost & risk
- Negotiations between suppliers for A and B brokered -> ICD

Conclusions

- We have looked at gaps in the literature, and started to overcome the lack of a lifecycle-oriented view of interface evolution.
- We have outlined some key principles associated with interfaces, and looked at some best practice methods of representing and elaborating them.
- We have stressed the use of interface analysis in architecting Systems throughout their lifecycle.
- We have encouraged engineers to look outside the box.





30th Annual INCOSE international symposium

Virtual Event

July 20 - 22, 2020

www.incose.org/symp2020