CS 428 WEBSTER #6 PART II

Winter 2022

Bruce F. Webster

POLITICAL PITFALLS

- Not educating and enlisting management before the fact
- Underestimating the resistance
- Overselling the technology or methodology
- Not recognizing the politics of architecture
- Getting religious about the technology or methodology
- Getting on the feature release treadmill
- Betting the company on a given technology or methodology
- Picking the wrong horse

NOT EDUCATING AND ENLISTING MANAGEMENT BEFORE THE FACT

- **Issue**: adopting a new technology or management (TOM) w/out notifying management of the potential issues ahead of time can backfire
- **Symptoms**: failure to meeting management expectations about the TOM
- **Consequences**: lack of support/backing from management if things go south
- **Detection**: sit down with management and find out what *they* think the TOM will do
- **Extraction**: better late then never -- make sure management has a realistic understanding and expectation for the TOM
- **Prevention**: first, make sure your own expectations are realistic; second, make sure management has its own expectations appropriately set

UNDERESTIMATING THE RESISTANCE

- **Issue**: not everyone is going to want your team/project/firm to adopt the new TOM; remember, people hate change
- **Symptoms**: lack of support at different levels; people distancing themselves from your decisions
- **Consequences**: project slows down or even fails, with corresponding consequences
- **Detection**: can be hard; start one-on-one talks to try to find what key people think
- **Extraction**: push ahead; mollify; redirect; or abandon
- **Prevention**: sound out key people ahead of time and make sure your support is solid

OVERSELLING THE TECHNOLOGY OR METHODOLOGY

- **Issue**: enthusiasm can lead you to focus only on the anticipated benefits of adopting the TOM and none of the potential or real pitfalls
- **Symptoms**: you have to keep making excuses and/or people start asking pointed questions
- **Consequences**: loss of trust from those who believed you; back-stabbing from those who didn't.
- **Detection**: list the promised benefits of the TOM and then give a realistic appraisal of the likelihood of achieving them
- **Extraction**: start readjusting expectations immediately; "take no small slips"
- **Prevention**: "underpromise and overdeliver"

NOT RECOGNIZING THE POLITICS OF ARCHITECTURE

- **Issue**: as an architect, you occupy a role of *political* leadership and need to enlist the support and enthusiasm of team members (and management) for your proposed architecture
- **Symptoms**: assuming that everyone will just go along with what you say. Finger-pointing at you when roadblocks appear.
- **Consequences**: hidden or overt team dissention, poor morale, lack of cooperation
- **Detection**: how much support do you have? What are the obstacles you're facing?
- **Extraction**: hard, because you have to sooth feeling and enroll team members after problems have arisen.
- **Prevention**: enroll support and feedback up front

GETTING RELIGIOUS ABOUT THE TECHNOLOGY OR METHODOLOGY-

- **Issue**: getting overly optimistic/enthusiastic about the TOM
- **Symptoms**: an almost blind faith in the virtues of the TO M and a blindness to its pitfalls and failings
- **Consequences**: heated arguments and an inability to be objective
- **Detection**: ask those skeptical of the TOM about their reasons
- **Extraction**: dial back your (or others') enthusiasm; be open to skepticism; leave yourself an escape route
- **Prevention**: thoroughly research criticism of the TOM ahead of time and bake that into whatever presentations you make

GETTING ON THE FEATURE RELEASE TREADMILL

- **Issue**: success in a version release often brings a new slew of feature requests instead of a chance to retire any technical debt and/or fix known defects
- **Symptoms**: constantly having to push tech debt/bug fixes into a future release and not the current one
- **Consequences**: the cost of adding features goes up, and the product becomes less stable
- **Detection**: detail tech debt/deferred bugs to management and see response
- **Extraction**: may be hard; have to fight to fix what's important
- **Prevention**: from the very start, emphasize the need to reduce technical debt and open defects, and why their have priority over new features

BETTING THE COMPANY ON A GIVEN TECHNOLOGY OR METHODOLOGY

- **Issue**: While there are no silver bullets, we sometimes act as if there are, particularly with some shiny new technology or methodology.
- **Symptoms**: unrealistic expectations on the part of upper management, marketing, developers.
- **Consequences**: major schedule adjustment or even project/market failure.
- **Detection**: does the current business plan make sense absent the TOM?
- **Extraction**: list the factors required for success independent of the TOM – is the project still feasible?
- **Prevention**: bet your company on your people, not a particular TOM.

PICKING THE WRONG HORSE

- **Issue**: sometimes we bet on a platform, environment, or TOM that doesn't pan out (NeXTstep, Windows Phone, etc.), due to overly optimistic expectation
- **Symptoms**: outside sources continue to raise issues and problems with your target platform/environment/TOM. Development resources hard to find.
- **Consequences**: your project struggles to get developers and potential customers. Even if the system ships, it may lack market size or ongoing support.
- **Detection**: talk frequently with other firms adopting the same TOM. Fact-check claims about potential market share and benefits.
- **Extraction**: migrate to a different platform or TOM.
- **Prevention**: this is hard, given the risk/reward of being the first to take advantage of a successful platform or TOM.