

# StranglerFigApplication

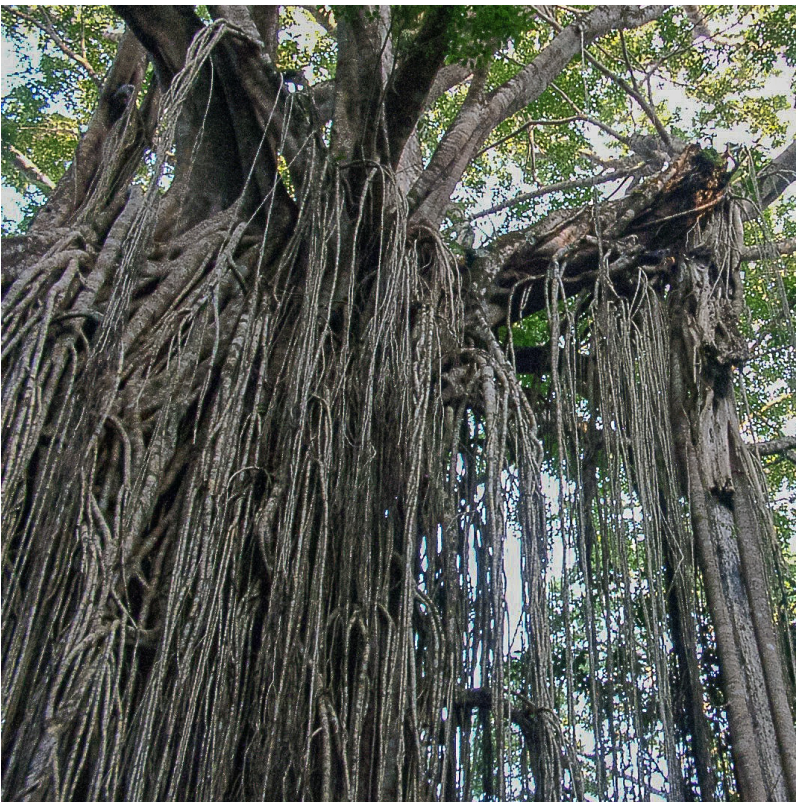
29 June 2004



**Martin Fowler**

APPLICATION ARCHITECTURE  
LEGACY REHAB

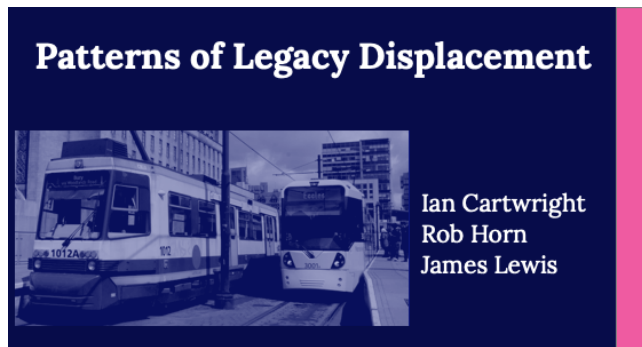
When Cindy and I went to Australia, we spent some time in the rain forests on the Queensland coast. One of the natural wonders of this area are the huge strangler figs. They seed in the upper branches of a tree and gradually work their way down the tree until they root in the soil. Over many years they grow into fantastic and beautiful shapes, meanwhile strangling and killing the tree that was their host.



This metaphor struck me as a way of describing a way of doing a rewrite of an important system. Much of my career has involved rewrites of critical systems. You

would think such a thing as easy - just make the new one do what the old one did. Yet they are always much more complex than they seem, and overflowing with risk. The big cut-over date looms, the pressure is on. While new features (there are always new features) are liked, old stuff has to remain. Even old bugs often need to be added to the rewritten system.

## Patterns of Legacy Displacement



This post has become very popular for a metaphor that was dashed off two decades ago, but the topic of replacing legacy systems deserves more considered attention. Fortunately my colleagues have stepped in to fill the gap, where they are working on a series of patterns to capture our experience replacing legacy systems in a gradual manner that allows new value to appear earlier with less risk.

**by Ian Cartwright, Rob Horn, ARTICLE  
and James Lewis 7 Jul 2022**

**Read more...**

**ENTERPRISE  
ARCHITECTURE EVOLUTIONARY  
DESIGN LEGACY REHAB**

An alternative route is to gradually create a new system around the edges of the old, letting it grow slowly over several years until the old system is strangled. Doing this sounds hard, but increasingly I think it's one of those things that isn't tried enough. In

particular I've noticed a couple of basic strategies that work well. The fundamental strategy is EventInterception, which can be used to gradually move functionality to the strangler fig and to enable AssetCapture.

My colleague Chris Stevenson was involved in a project that did this recently with a great deal of success. They published a first paper on this at XP 2004, and I'm hoping for more that describe more aspects of this project. They aren't yet at the point where the old application is strangled - but they've delivered valuable functionality to the business that gives the team the credibility to go further. And even if they stop now, they have a huge return on investment - which is more than many cut-over rewrites achieve.

The most important reason to consider a strangler fig application over a cut-over rewrite is reduced risk. A strangler fig can give value steadily and the frequent releases allow you to monitor its progress more carefully. Many people still don't consider a strangler fig since they think it will cost more - I'm not convinced about that. Since you can use shorter release cycles with a strangler fig you can avoid a lot of the unnecessary features that cut over rewrites often generate.

There's another important idea here - when designing a new application you should design it in such a way as to make it easier for it to be strangled in the future. Let's face it, all we are doing is writing tomorrow's legacy software today. By making it easy to add a strangler fig in the future, you are enabling the graceful fading away of today's work.

## Further Reading

Paul Hammant has a good summary of case studies using this approach.

## Revisions

Changed URL and name to Strangler Fig Application April 29 2019

I thought this post was a nice metaphor, but didn't expect the degree that it would grow in popularity (in recent months it gets over 3000 page views a month). The popularity is nice, but there is a problem. The original post was entitled “Strangler Application”, and when used, the pattern is often referred to as a “strangler”. But its usage often gets separated from its metaphorical root, and takes on a unpleasantly violent connotation.

Some people, therefore, have advocated avoiding or changing the name. I don't have any great objection to that, I haven't used the name in my own writing since that original posting. But the trouble with attempting a rename, is that once a name has lodged itself in a community's mind, it's very hard to dislodge.

Recently I thought of a small tweak that might help things a little. If I rename the post to “Strangler *Fig* Application”, and use the term “Strangler Fig” as much as possible, then hopefully that would reduce the violent connotation by reinforcing the metaphorical link that is the whole point of the name. Because it's a small change, maybe it will spread enough to be worthwhile, and it's not much effort, so seems worth a try.