Pivotal Summit 2019

Pivotal’s approach

To Strangle or Not to Strangle

Siva Manickavasagam
Director, Application Transformation
Asia Pacific and Japan
Pivotal Software
**Strangle**

<table>
<thead>
<tr>
<th>Definition</th>
<th>Pronunciation</th>
<th>Derived Forms</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UK</td>
<td></td>
<td>Past Participle: strangled  Present Participle: strangling</td>
</tr>
</tbody>
</table>

Unless otherwise indicated, these slides are © 2013-2019 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: http://creativecommons.org/licenses/by-nc/3.0/
The Strangler Application / Pattern

What (어떤 애플리케이션을?)

When (어느 시점에?)

How (어떤 방법으로?)
Strangler fig, also called strangler, any of numerous species of tropical figs (genus Ficus, family Moraceae) named for their pattern of growth upon host trees, which often results in the host's death. Strangler figs and other strangler species are common in tropical forests throughout the world.

strangler fig | Definition, Life Cycle, & Species | Britannica
https://www.britannica.com › plant › strangler-fig-tree
무화과
무화과 나무가 숙주나무에서 자라기 시작

숙주 위에서 서서히 성장하며 울아omap
결국 숙주나무는 죽었다
Roads = Tech Apps
도로 = 애플리케이션
Highways = Monoliths
고속도로 = 모놀리스 앱
Monolith

StranglerA

ロード 뱅커
How to select Monoliths for Strangling?

1. Portfolio
2. Apply Funnel Intake Filter
3. Candidates
4. Assess Business Value
5. Business Value
6. Assess Tech Feasibility
7. Tech vs. BV
8. Tech Prioritization
9. 4 Quadrants
10. Export to Backlog
11. Tracker Backlog
12. Apply Program Guidance
13. Pivotal

Program Plan

current

icebox
When to Strangle?

- RePlatform
- ReFactor
- ReHost / Retire
- Rewrite (Strangle / Others)
When to Strangle?

**Critical Mass** - Code, Throughput, User base

**Age** - Secrets of the Wily Old App!
Strangling Monoliths - The SWIFT way

1. Define Objectives & Key Results
(목표 및 주요 결과 정의하기)

2. Event Storm the legacy space
(기존 레거시에 대한 이벤트 스토밍)

3. Select “Thin Slice(s)”
(대상 슬라이스 선택하기)

4. “Boris” crafts the desired notional architecture
(Boris를 통한 개념적 아키텍처 그리기)

5. Snap-E
(각 서비스 정의하기)

6. Produce tested and working code
(동작하는 코드 작성하기)

7. Cloud Modernization Patterns
(전환 패턴 찾아내서 정리하기)
Approach to the Strangler needs to be holistic

**Culture**
Continually improve and deliver customer success

**Methodology**
Learn the necessary practices to build modern software

**Tools**
Build with products focused on developer productivity

**Platform**
Run every app, on every cloud, on a unified platform

**Cloud-native software**
The Strangler Application / Pattern

**What** (어떤 애플리케이션을?)

**When** (어느 시점에?)

**How** (어떤 방법으로?)
Thank You

Email: smanickavasagam@pivotal.io
LinkedIn: https://www.linkedin.com/in/SivaManickavasagam

We Love Feedback
What would you like to hear more about?
What questions do you still have?

Recommended Viewing / Reading

• Event Storming: https://www.eventstorming.com/
• App Continuum: https://www.appcontinuum.io/
• Data Strangler: https://www.infoq.com/presentations/monolith-refactoring-data-driven-testing/