Strategic Prototyping for Developing Big Data Systems

Hong-Mei Chen, University of Hawaii
Serhiy Haziyev, SoftServe Inc.
Olha Hrytsay, SoftServe Inc.
Rick Kazman, University of Hawaii and SEI/CMU
Agenda

Small & Big Data Systems
Types of Prototypes
RASP Model
Discussion
Small (Traditional) vs Big Data Systems

**Traditional Analytics (BI)**
- Reporting/BI
- OLAP Cubes
- Data Warehouses
- ETL
- Structured Data

**Big Data Analytics**
- Reporting/BI
- Machine Learning
- MPP Data Warehouses
- Hadoop/Map Reduce
- NoSQL Databases
- Search Engines
- Structured Data
- Semi-Structured Data
- Unstructured Data
Big Data Systems Challenges

- The five V’s of big data (Volume, Variety, Velocity, Veracity, and Value)
- Paradigm shifts (Data Lake, Polyglot Persistence, Lambda Architecture, etc.)
- The short history of big data system development
- Rapid technology changes
- The difficulty of selecting big data technology
Agenda

- Small & Big Data Systems
- Types of Prototypes
- RASP Model
- Discussion
Why Prototyping is Important?

Typical motivations for prototyping:
• Requirements are uncertain
• Technologies are new
• No comparable system has been previously developed
• No full buy-in from the business

Architectural analysis alone can not address these risks!
Types of Prototypes

Horizontal Prototype

Vertical Evolutionary Prototype

Throwaway Prototype (Proof-of-Concept)

Minimum Viable Product (MVP)
When and Why to Prototype?

Goals (Why?):

- Identification of missing, conflicting or ambiguous architectural requirements
- Creation of initial architecture design and selection of candidate technologies
- Confirmation of user interface requirements and system scope
- Demonstration version of the system to obtain buy-in from the business
- Validation of technologies and scenarios that pose risks
- Integration of selected technologies
- Clarification of complex requirements
- Testing critical functionality and quality attribute scenarios
- Getting early feedback from end users and updating the product accordingly
- Presentation of a working version to a trade show or customer event
- Evaluation of team progress and alignment

Agenda

- Small & Big Data Systems
- Types of Prototypes
- RASP Model
- Discussion
The RASP Model

We created *RASP* (Risk-based Architecture-centric Strategic Prototyping) to provide guidance on how to employ strategic prototyping for risk management.

The RASP model can stand alone, but is ideally integrated with an architecture-centric agile development methodology, such as BDD (Big Data Design).
BDD and RASP

**Value Discovery**
- Innovation process
- Use case development
- Strategic development planning

**Requirements Analysis**
1. Business and innovation goals
2. Constraints and concerns
3. Quality attribute scenarios
4. Big data architecture scenarios

**Design**
1. Choose a reference architecture
2. Form the architecture landscape
3. Establish an iteration goal
4. Choose the elements to decompose
5. Choose design concepts and data models
6. Instantiate the architectural elements
7. Sketch views; record decisions and metadata
8. Evaluate each iteration
9. Architecture analysis: BITAM

**Implementation and Testing**

**Evaluation:**
- Technical and business dimensions

**BDD Architecture Analysis**
- Identify the risks in business goals, alignment, and strategy

**RASP Architecture Analysis**
- Identify the risks of missing, conflicting, ambiguous, or underspecified architectural requirements

**RASP Strategic Prototyping**
- Feasibility study and UI mockups (throwaway prototyping)

- Identify the risks for each decision made (or not made) related to high-priority scenarios and alignment with business goals

- Technology risk mitigation (vertical evolutionary and throwaway prototyping)

- Minimum viable product (MVP)
RASP Model: A strategic-prototyping decision flowchart

**Initial Architecture Analysis**
- Assess existing system and identify gaps that pose risks for new requirements
- Is this a green field system design?
  - Yes: Perform architecture analysis, and select a reference architecture and candidate technologies
  - No: Perform architecture analysis, update the existing architecture, and select candidate technologies to replace or augment existing technologies

**Evolutionary Prototype**
- Go to throwaway prototype chart for each technology that poses a risk
- Is there a significant risk with a selected technology?
  - Yes: Integrate technologies, test critical functionality and quality attribute scenarios
  - No: Are there issues discovered with any technologies?
    - Yes: Reconsider the technology selection, and go to throwaway prototype chart
    - No:
RASP Model: A strategic-prototyping decision flowchart

1. **Is the selected technology new on the market?**
   - Yes: **Change candidate technology**
   - No:
     - **Are there new requirements that pose technology risks?**
       - Yes: Create rapid (throwaway) prototype to mitigate risks
       - No: Proceed to the next decision
     - No:
       - **Was it used successfully before for similar scenarios?**
         - Yes: Proceed to the next decision
         - No: **Have trusted sources “proven” the technology for specific scenarios?**
           - Yes: Proceed to the next decision
           - No: **Create rapid (throwaway) prototype to mitigate risks**
     - **Did the prototype satisfy the specific scenarios?**
       - Yes: Proceed to the next decision
       - No: **Are there more configurations, data models, resources to try?**
         - Yes: Proceed to the next decision
         - No: **Is it possible to “soften” any requirements?**
           - Yes: Proceed to the next decision
           - No: Proceed to the next decision

2. **Are there new requirements that pose technology risks?**
   - Yes: Create rapid (throwaway) prototype to mitigate risks
   - No: Proceed to the next decision

3. **Was it used successfully before for similar scenarios?**
   - Yes: Proceed to the next decision
   - No: Proceed to the next decision

4. **Have trusted sources “proven” the technology for specific scenarios?**
   - Yes: Proceed to the next decision
   - No: Proceed to the next decision

5. **Create rapid (throwaway) prototype to mitigate risks**
   - Yes: Proceed to the next decision
   - No: Proceed to the next decision

6. **Does the prototype satisfy the specific scenarios?**
   - Yes: Proceed to the next decision
   - No: Proceed to the next decision

7. **Are there more configurations, data models, resources to try?**
   - Yes: Proceed to the next decision
   - No: Proceed to the next decision

8. **Is it possible to “soften” any requirements?**
   - Yes: Proceed to the next decision
   - No: Proceed to the next decision
Validation

9 case studies were used to validate RASP


Diverse BD requirements: data volume, latency, scalability

Details available at: http://itm-vm.shidler.hawaii.edu/BDCases/CaseStudies.htm
Summary: 7 Guidelines

1. Architecture analysis might be the only feasible option when you need to make early decisions
2. Architecture analysis alone is insufficient to prove many important system properties
3. Architecture analysis complements vertical evolutionary prototyping
4. An evolutionary prototype can effectively mitigate risk if you implement it as a skeleton
5. Vertical evolutionary prototyping might need to be augmented with throwaway prototypes
6. Throwaway prototypes work best when you need to quickly evaluate a technology
7. MVP is more of a business decision than a decision driven simply by technological risk
Agenda

1. Small & Big Data Systems
2. Types of Prototypes
3. RASP Model
4. Discussion
Conclusions

Risk analysis has been advocated for controlling the evolutionary prototyping process.

Our research results demonstrate that an architecture-centric design methodology such as BDD makes risk management explicit, systematic and cost-effective.

It provides a basis for value discovery with stakeholders, for reasoning about risks, for planning and estimating cost and schedule, and for supporting experimentation.

The RASP model offers practical guidelines for strategic prototyping, combining architecture analysis and a variety of prototyping techniques.