# Goal-Question-Metric Example

## GQM Goal 1

Evaluate the effect of Javascript client frameworks on ease of development from the point of view of developers in the context of a company using a .Net environment.

Purpose: evaluate

Object of study: Javascript client frameworks (Knockout, Angular, Ember)

Focus: ease of development

Stakeholder: developers

Context: a company using a .Net environment

### Questions and Metrics for GQM Goal 1

Q1: How easy is it to write a listing application using the API of this JS framework?

Metrics 1: easy/medium/hard/not possible to adapt (categorical)

Strategy 1: Adapt the given application using this framework (and a smaller number of records, like 10)

Q2: How much up-to-date API documentation is available?

Metrics 2: count of documented services (quantitative)

Strategy 2: Identify how many services are documented in the API.

Q3: Is there a ‘getting started’ tutorial or sample application?

Metrics 3: number of tutorials/samples (quantitative)

Strategy 3: Search online.

Q4: What design pattern does it implement? (MVC, MVVM, MVT, MTC, etc.)

Metrics 4: which design pattern (categorical)

Strategy 4: Code inspection, accompanied by framework creator’s explanation of the design’s benefits.

## GQM Goal 2

Evaluate the effect of Javascript client frameworks on scalability and performance from the point of view of software architects in the context of a company using a .Net environment.

Purpose: evaluate

Object of study: Javascript client frameworks (Knockout, Angular, Ember)

Focus: scalability and performance

Stakeholder: software architects

Context: a company using a .Net environment

### Questions and Metrics for GQM Goal 2

Q5: How many records can the framework handle?

Metrics 5a: max number of records it can load (quantitative)

Strategy 1 again (adapt the application)

Strategy 5a: keep adding more records (1,000; 10,000; 100,000; eventually 500,000) until application crashes or hangs

Metrics 5b: measurement of memory use (quantitative)

Strategy 5b: measure memory utilization using system monitor

Q6: How fast does it load? (visible load time and total load time)

Metrics 6a: visible load time (quantitative)

Strategy 6a: measure time from page request to data starting to display

Metrics 6b: total load time (quantitative)

Strategy 6b: measure time from page request to data finishing loading

Q7: Which browser is faster?

Metrics 6a and 6b again

Strategy 7: repeat Strategy 6a and 6b for Chrome, Firefox, IE