

Process & Project Metrics

Metrics Philosophy:

Spiral is a methodology in which it is uniquely important to have informative metrics. The reason for this lies in its reliance on risk analysis; given this is a research project and there are points at which we may start down roads that lead nowhere, roadmapping good software metrics is imperative, as it directly lends to more accurate risk analysis. This allows us to more confidently evaluate our process; how it has worked in the past, and how it should work in the sprint we are planning for the future.

Based on this, we are planning to implement a robust project metric plan. After every sprint in the Spiral, we will have a metric planning and review meeting. In this meeting, metrics will be reviewed for the previous sprint. Each metric will be evaluated with a score from 1-100 and separated into categories based on whether they are a software or process metric. This score will depend on several factors unique to each, some data will come from external sources, and some will come from internal testing. An overall score will be assigned for each section (Process vs. Software Functionality) and a review will be conducted to determine our overall progress in the past sprint. This will help us to determine any problem areas we are specifically struggling with, and allow us to have better risk analysis and planning in the future.

Process Metrics:

1. Bug Resolution Time

- Track, on average, the time it takes for bugs in the backlog to be fixed
 - This will give us more insight to help with risk analysis, an important part of the Spiral methodology

2. Code Churn

- Track the quality of our code based on frequency of reworks
 - Helpful metric for planning sprints and risk analysis

3. Velocity

- Track the average work we get done per sprint

4. Defect Density

- Track the number of errors we find in any artifact

5. Working Hours

- Track the number of hours allocated working per member, per week.

Software Functionality Metrics:

1. Camera Shot Quality

- Outside observers compare robot's camera movements to a regular operator's
 - Rates cinematography
 - Rates aesthetics

2. Predictive Accuracy

- Measurement of how accurate our computer vision predictive model is

3. Success of Target Lock

- Track how often the computer vision algorithm loses its target

4. End User Experience

- Have volunteers with a feasible use-case test the arm and evaluate it based on:
 - ease of use
 - practicality
 - whether or not they would use it day-to-day

5. **Camera -> Arm Latency**

- Measure how long it takes for camera readings to pass to the arm and for the arm to take action