CMMI® Institute

THE HIGH MATURITY JOURNEY OF NEXT GENERATION DESKTOP DEVELOPMENT

NAME: Steve Moulder

TITLE: Process Engineer Advisor

ORGANIZATION: National Government Services

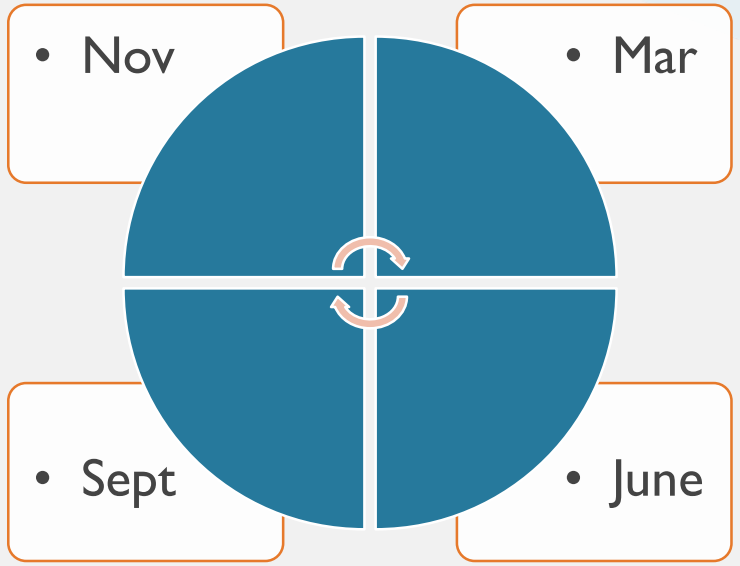
National Government Services℠

NEXT GENERATION DESKTOP BACKGROUND

- Next Generation Desktop (NGD) developed by National Government Services (NGS), an Anthem company
- NGD Supports call centers for the Centers for Medicare & Medicaid Services (CMS)

NEXT GENERATION DESKTOP BACKGROUND

- NGD Issues Quarterly Releases
- Releases follow a full Lifecycle based on the CMS Expedited Lifecycle (XLC)



NGD OBJECTIVES

CRITICAL BUSINESS OBJECTIVE

- Improve Quality
  - Increase First Time Quality (FTQ)
  - Also means Reduce Rework
  - Reduce rate of Defect Injection

QUALITY AND PROCESS PERFORMANCE OBJECTIVES

- Achieve and maintain average First Time Quality in Test greater than or equal to 90%
- Achieve and maintain average First Time Quality in UAT greater than or equal to 95%

INITIAL FIRST TIME QUALITY DEFINITION

- FTQ =

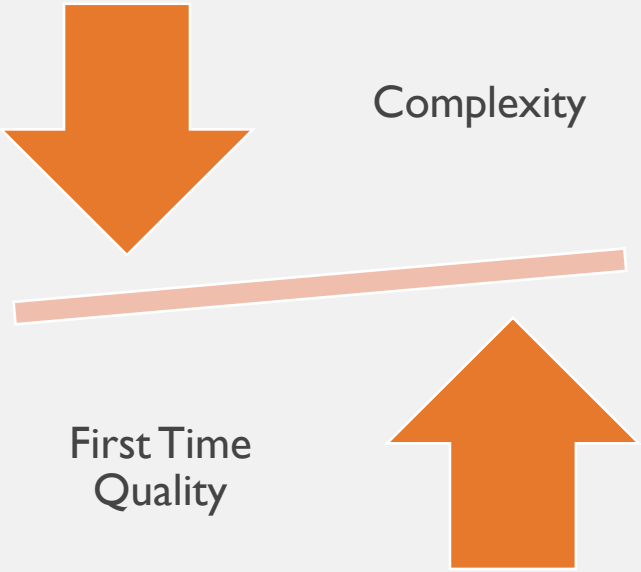
  (1 - (number of tasks with defects) / (Total Number of Tasks)) * 100%

  Another way of stating this

- FTQ = 100% - Percent Defective Tasks

HYPOTHESIS

- First Time Quality is Inversely proportional to Program Complexity



COMPLEXITY

- Program Complexity Index (PCI)

  = (3 * X) + (2 *Y) + (1 * Z)

Where

X = High Complexity Effort

Y = Medium Complexity Effort

Z = Low Complexity Effort

Complexity Legend		Proposed Actions
	3.0	• Decompose high complexity work items
	2.9	• Consider deferring a subset of high complexity work items
	2.8	or adding another development drop
	2.7	• Hold internal requirements reviews with the team
	2.6	
	2.5	
	2.4	
	2.3	
	2.2	
UCL	2.1	• Hold additional peer and drop reviews
	2.0	• Expand unit test coverage
	1.9	
	1.8	
	1.7	• No adjustments needed
	1.6	
Mean	1.5	
	1.4	
	1.3	
	1.2	• Consider adding enhancement work items to the project
	1.1	
LCL	1.0	

EARLY ANALYSIS

- Observation:
  - There is a mismatch between First Time Quality and Program Complexity Index
    - FTQ relates to Number of Defective Tasks
    - PCI relates to estimated effort for each complexity level
- Adjustment:
  - Change Definition of First Time Quality
    - FTQ = (1 – (Rework / (First Time Effort))*100%
    - Where First Time Effort – Initial Development Effort + Initial Testing Effort

FURTHER ANALYSIS AND CAR

- Observation / Problem:
  - Data collected over 14 Releases shows no Statistically Significant Relationship between Program Complexity Index and First Time Quality

THE CAR

ROOT CAUSES

- Rules for assigning Complexity are vague and subjective
- Coefficients in the formula for calculating PCI

  (PCI = (X*3)+(Y*2)+(Z*1))

  were arbitrarily established with no statistical basis

CAR CONSIDERATIONS

POSSIBLE CORRECTIVE ACTIONS

1. Improve the PCI Model
   - Improve the rules for assigning Complexity
   - Collect data
   - Perform Regression Analysis
   - Update the Model
2. Replace the PCI Model with a different model

EVALUATION CRITERIA

- Effort required to implement the option
- Ability to utilize existing data
- Elapsed time required to implement the option and see results
- Estimated probability of success

CAR CONCLUSIONS

- Given that the Release Cycle is Quarterly, it would take a few years to implement option 1
- The team estimated a higher probability of success for option 2
- Decision:
  - NGD would examine other factors, looking for a statistically significant relationship on which to build a new Predictive Process Performance Model

DEVELOPING A NEW MODEL

- A Statistically Significant Relationship was discovered between Effort per Development Task and First Time Quality
- There is also a Statistically Significant Relationship between Total Effort in a Release and Number of Defects

CONCEPTUAL MODELS

- First Time Quality

  $= C + A*X + B*X^2$

- Where
  - X is First Pass Development and QA Effort
    - (First Pass Development and QA Effort is the initial effort to develop and test. It excludes Rework.)
  - A is a statistically determined coefficient for the linear term
  - B is a statistically determined coefficient for the quadratic term
  - C is a constant determined by regression analysis of data for a specific project

- Number of Defects

  = F + (E * Release Effort)

- Where
  - E is a statistically determined coefficient for the linear term
  - F is a constant determined by regression analysis of data for a specific project

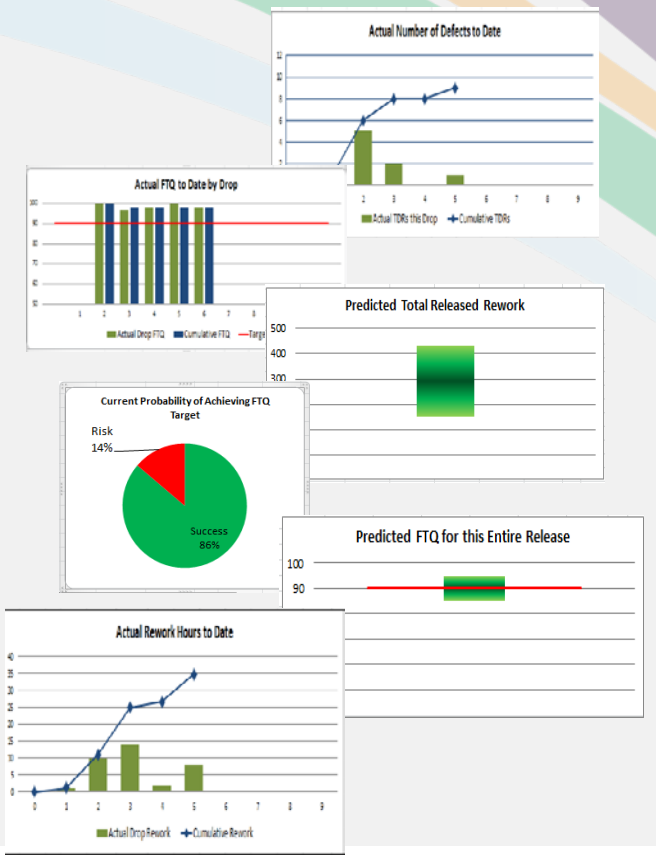
NGD FIRST TIME QUALITY MODEL

Input Data

- Total Estimated Effort
- Total Number of Dev Tasks
- Effort Completed
- Number of Tasks Completed
- Rework Completed
- Number of Actual Defects to Date
- Estimated Remaining Effort
- Number of Remaining Tasks

NGD FTQ Formula

Outputs

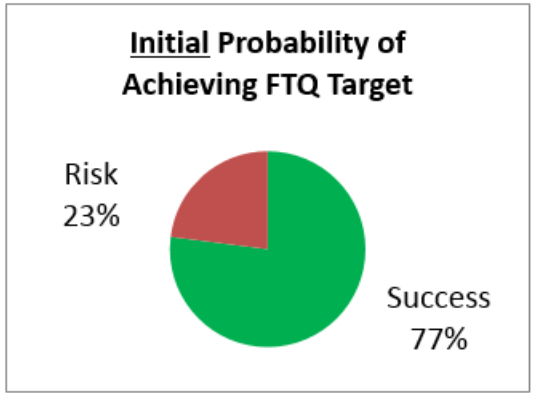


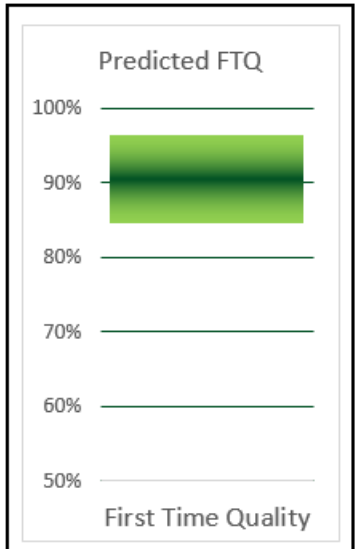
FTQ MODEL FEATURES

- Used at different points in the project life cycle including:
  - During Release Scope Planning
  - During "Drop" Planning
  - During Release Execution
- Supports "What-If" analysis
- Facilitates course corrections for "in-flight" Releases

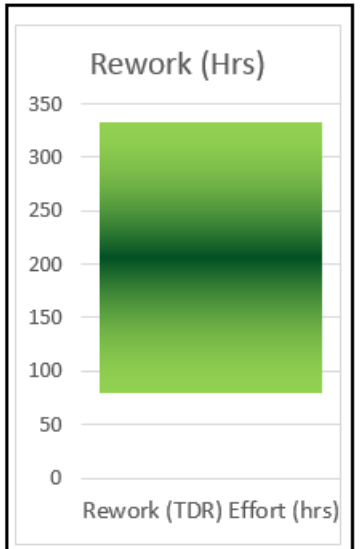
RELEASE SCOPE PLANNING

Initial Inputs	
Estimated Effort	2169
Number of Tasks	72

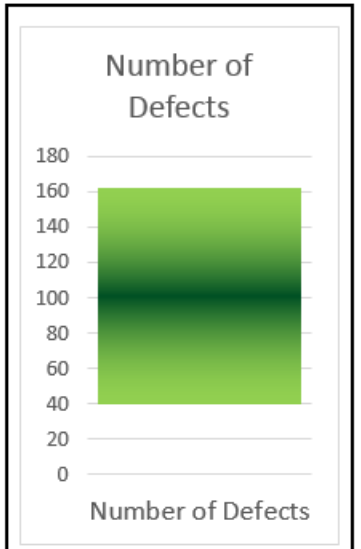




Upper Limit	96%
Most Probable	90%
Lower Limit	85%

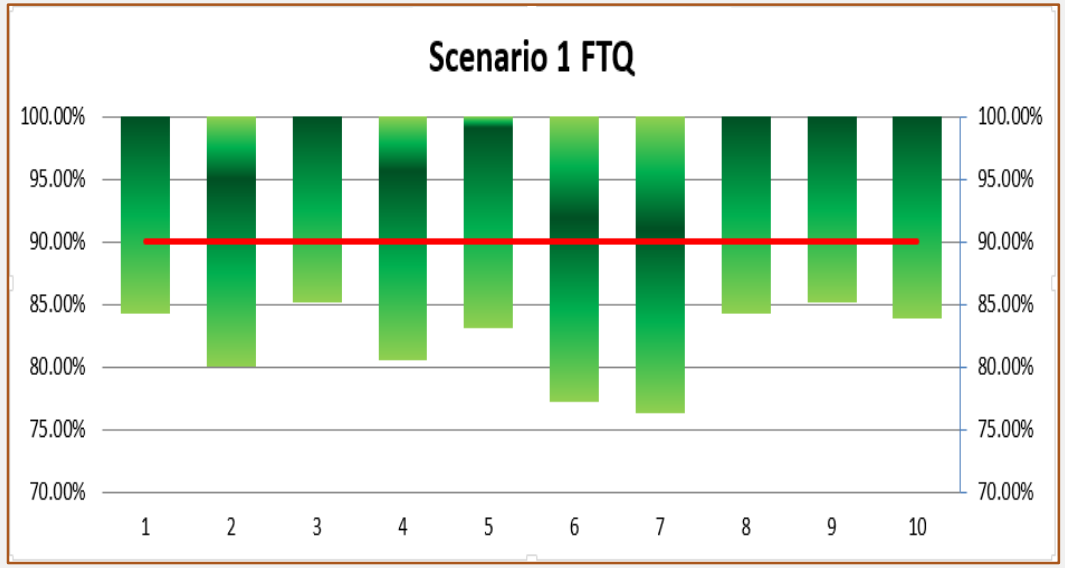


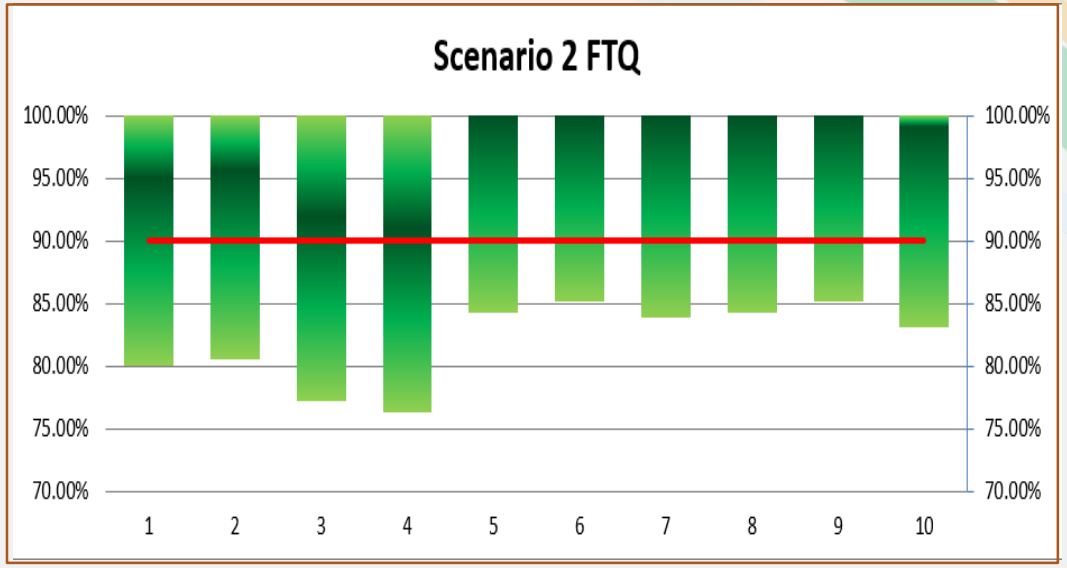
Upper Limit	333 hrs
Most Probable	206 hrs
Lower Limit	80 hrs



Upper Limit	162
Most Probable	101
Lower Limit	40

SCOPE AND DROP PLANNING

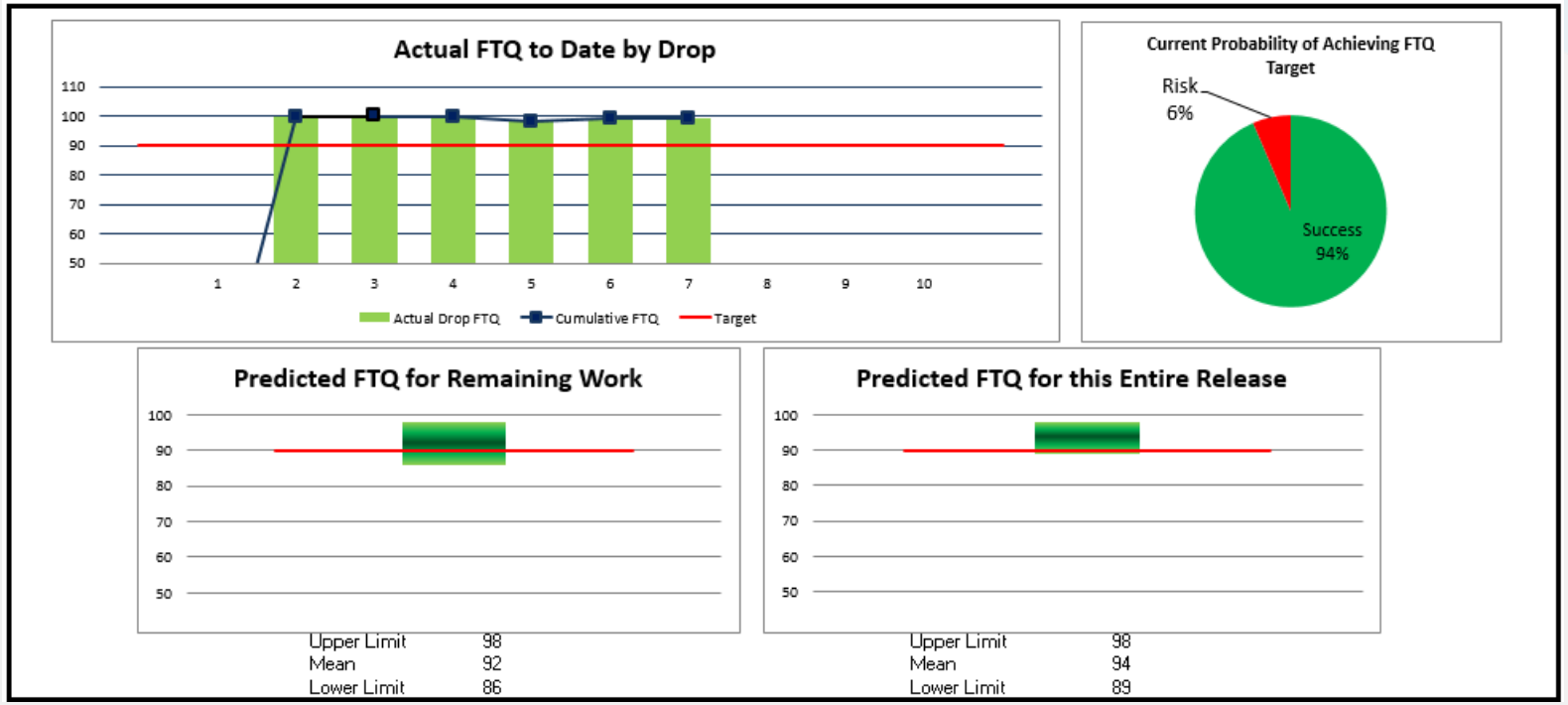




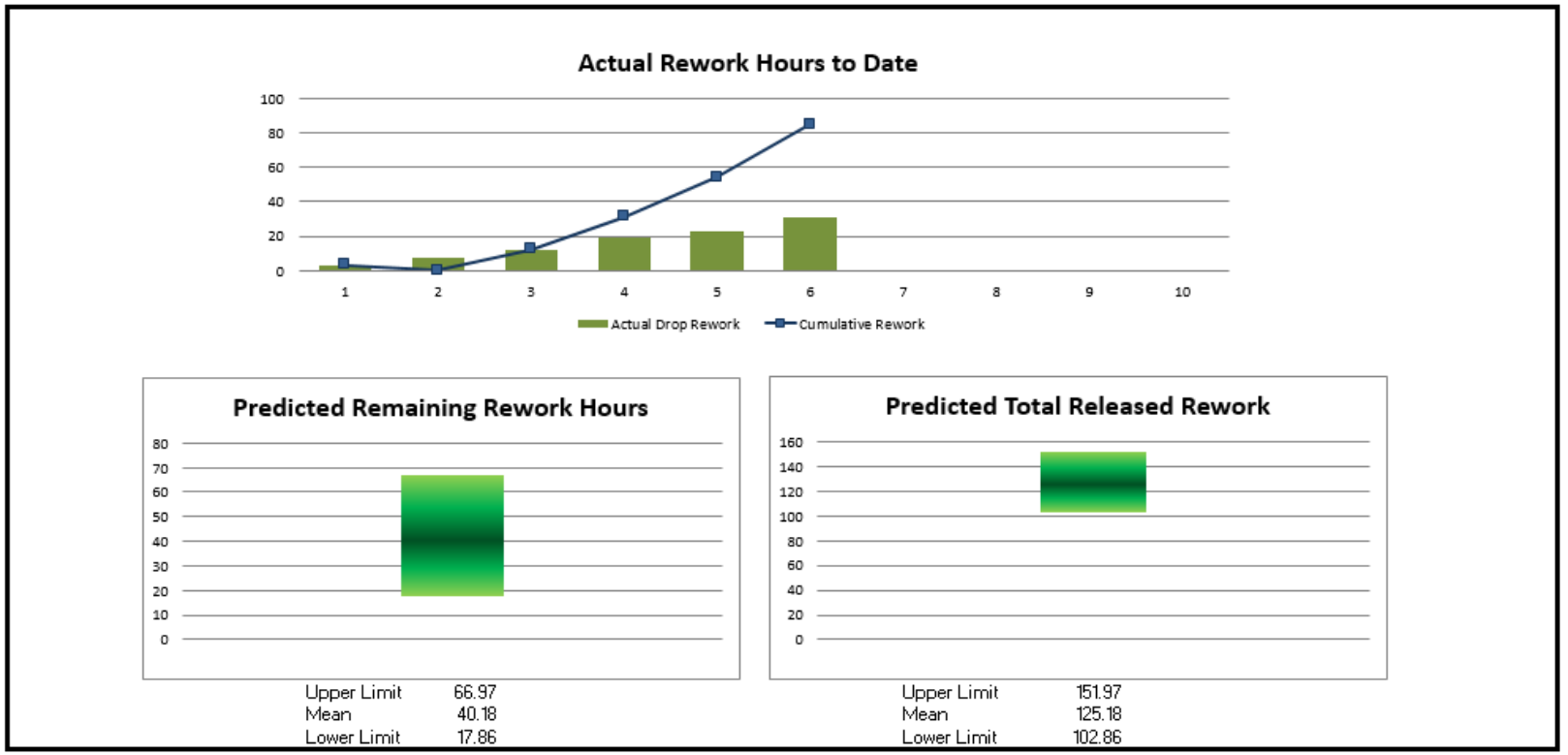
INPUTS DURING PROJECT EXECUTION

Ongoing Inputs	Drop	1	2	3	4	5	6	7	8	9	10
	Development Effort Completed	26.8	133.9	428.6	616.1	1098.3	1607.3	1875.2			
	QA Effort Completed	17.9	89.3	285.7	410.7	732.2	1071.5	1250.1			
	Number of Tasks Completed	1	5	16	23	41	60	70			
	TDR Effort (Rework) Complete	3	7	12	19	23	31	45			
	Number of Actual TDRs	5	12	22	40	63	79	90			
	Estimated Remaining DevEffo	1848.35	1741.25	1446.55	1259.05	776.85	267.85	0			
	Estimated Remaining QA Effo	1232.2	1160.8	964.4	839.4	517.9	178.6	0			
	Number of Tasks Remaining	69	65	54	47	29	10	0			

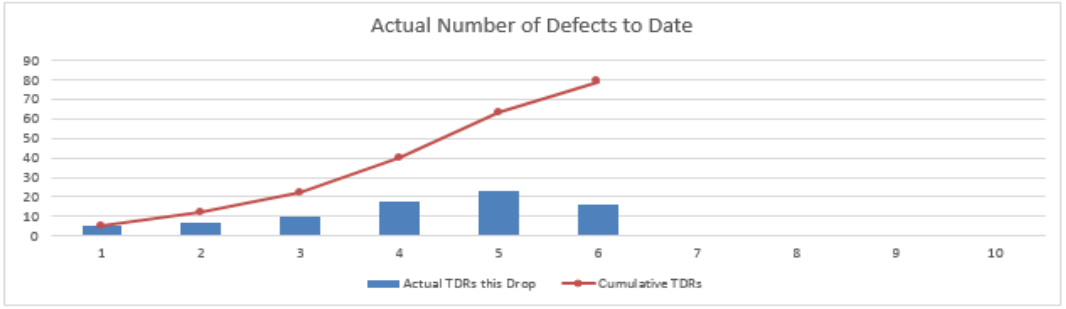
OUTPUTS DURING PROJECT EXECUTION
FIRST TIME QUALITY

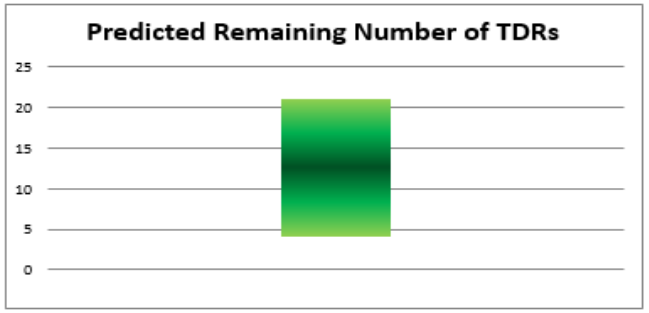


OUTPUTS DURING PROJECT EXECUTION REWORK

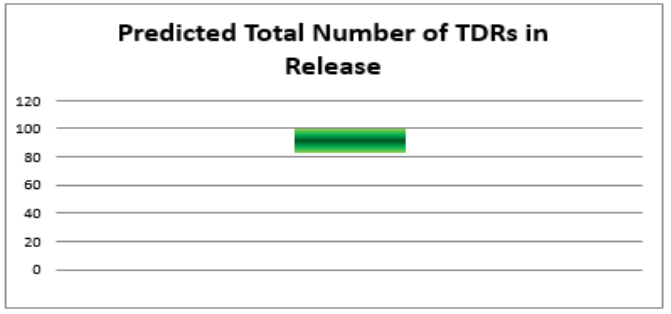


OUTPUTS DURING PROJECT EXECUTION
NUMBER OF DEFECTS



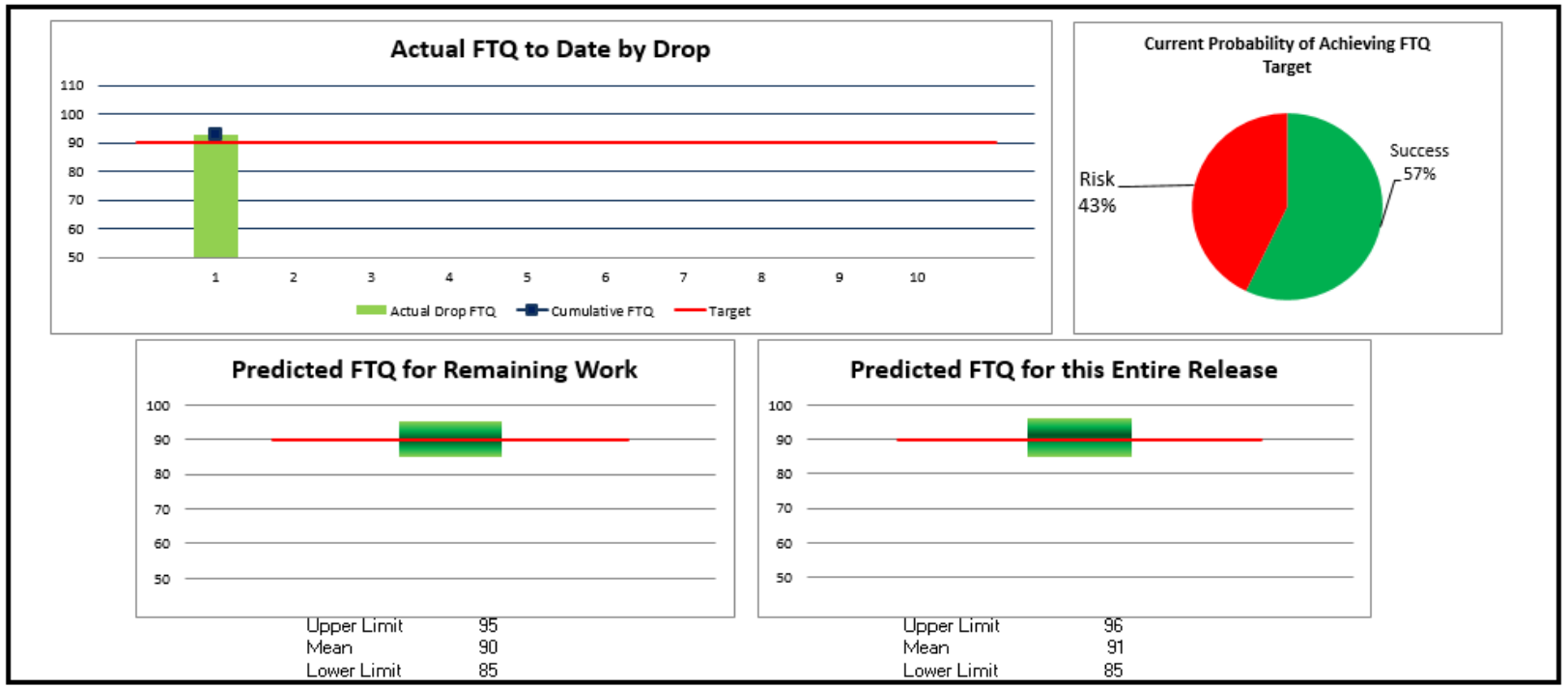


Upper Limit	21
Mean	13
Lower Limit	4

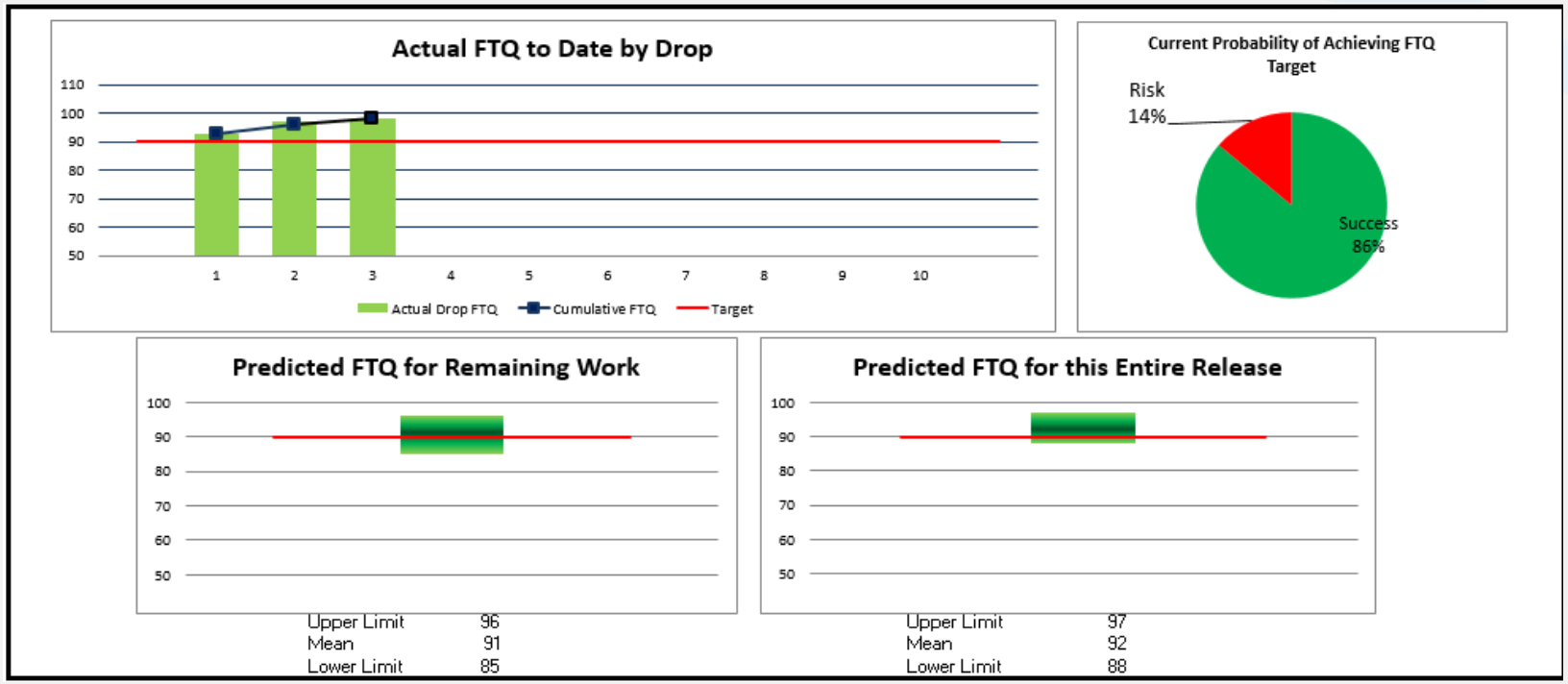


Upper Limit	100
Mean	92
Lower Limit	83

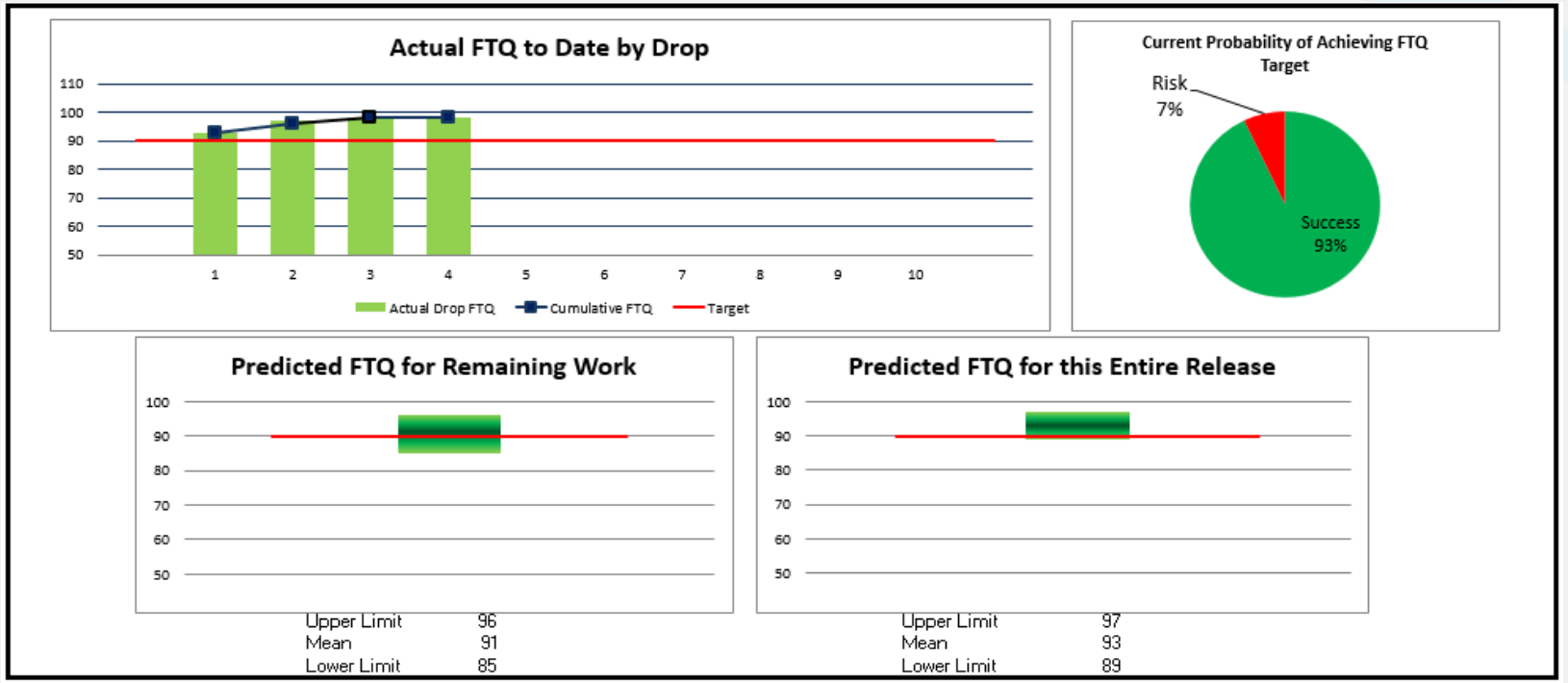
THE MODEL IN ACTION DURING PROJECT EXECUTION
FIRST TIME QUALITY – WEEK 1



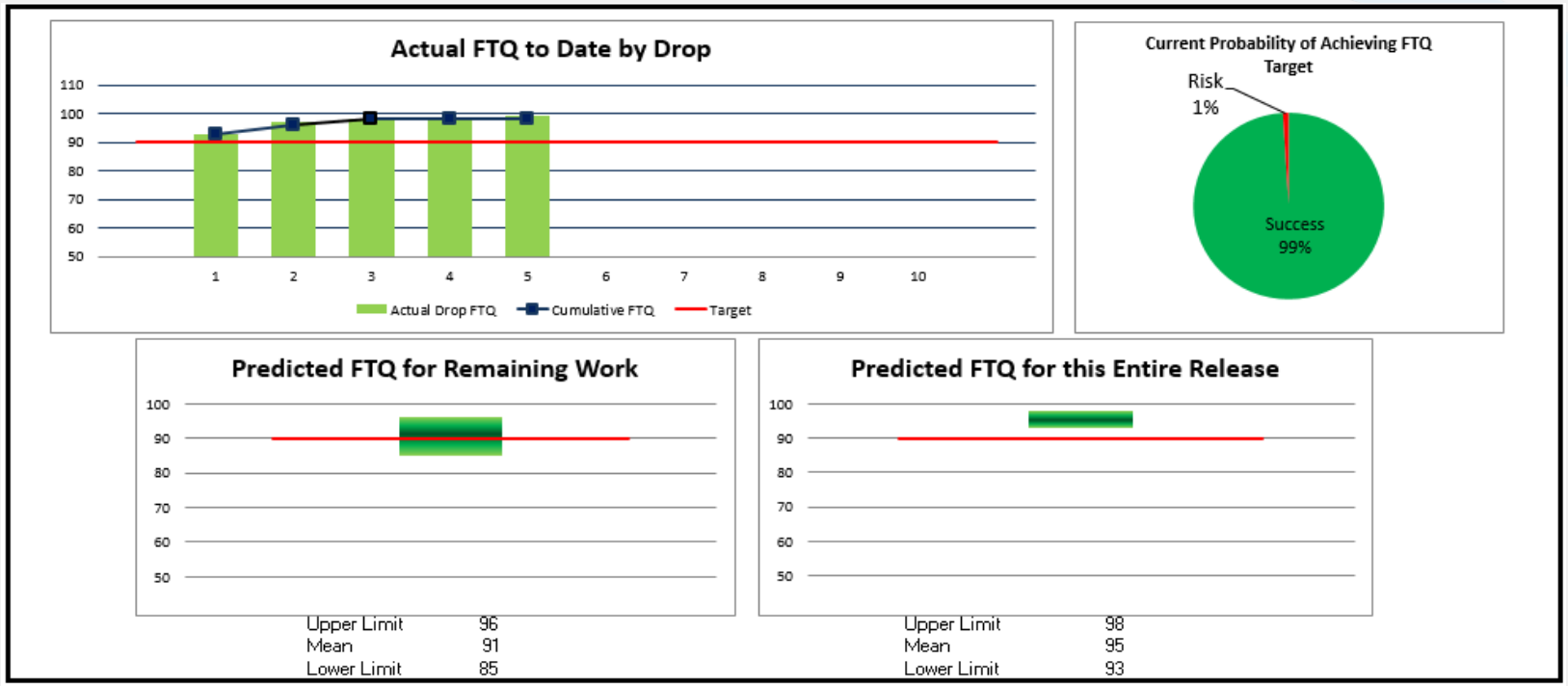
THE MODEL IN ACTION DURING PROJECT EXECUTION
FIRST TIME QUALITY – WEEK 3



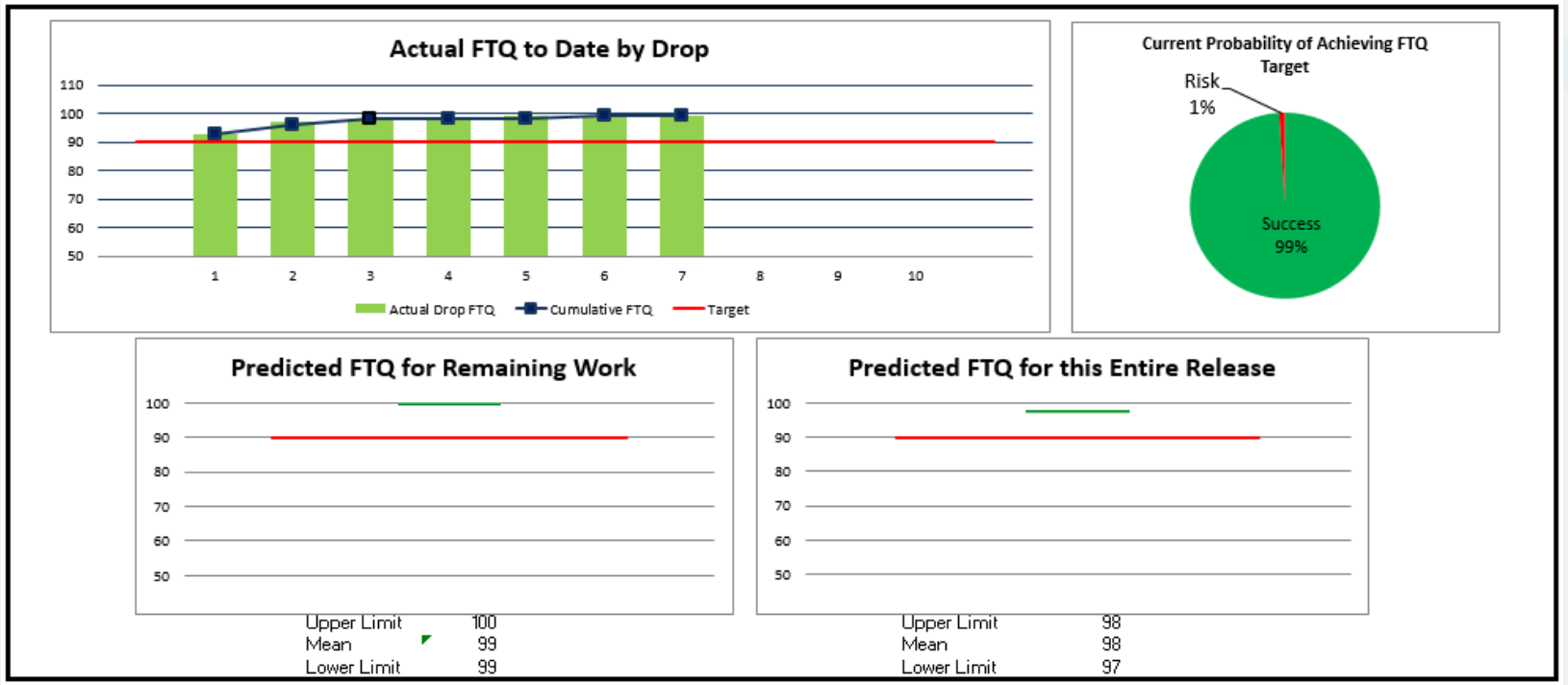
THE MODEL IN ACTION DURING PROJECT EXECUTION
FIRST TIME QUALITY – WEEK 4



THE MODEL IN ACTION DURING PROJECT EXECUTION
FIRST TIME QUALITY – WEEK 5

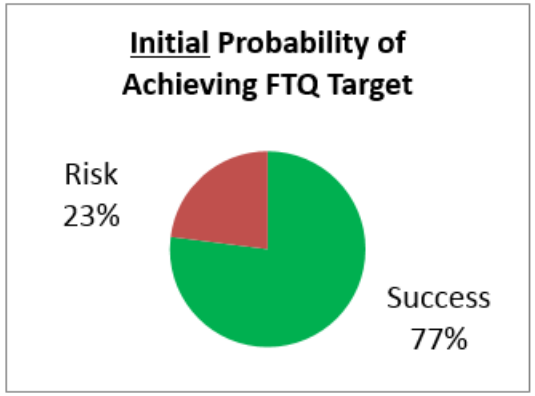


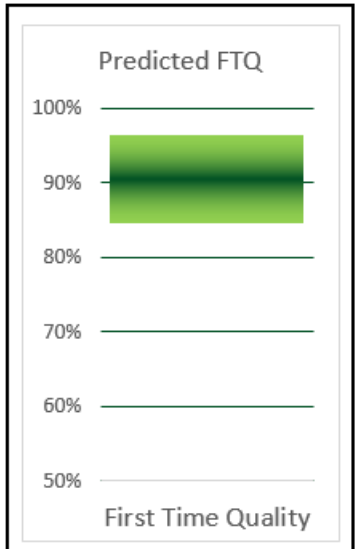
THE MODEL IN ACTION DURING PROJECT EXECUTION
FIRST TIME QUALITY – WEEK 7



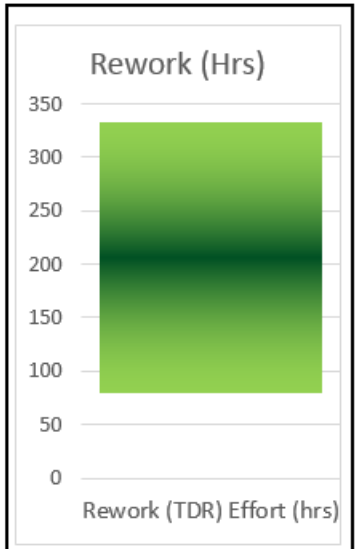
MODEL CONTINUOUS IMPROVEMENT RELEASE SCOPE PLANNING

Initial Inputs	
Estimated Effort	2169
Number of Tasks	72

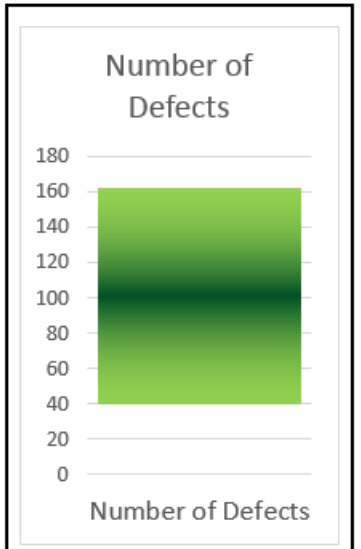




Upper Limit	96%
Most Probable	90%
Lower Limit	85%

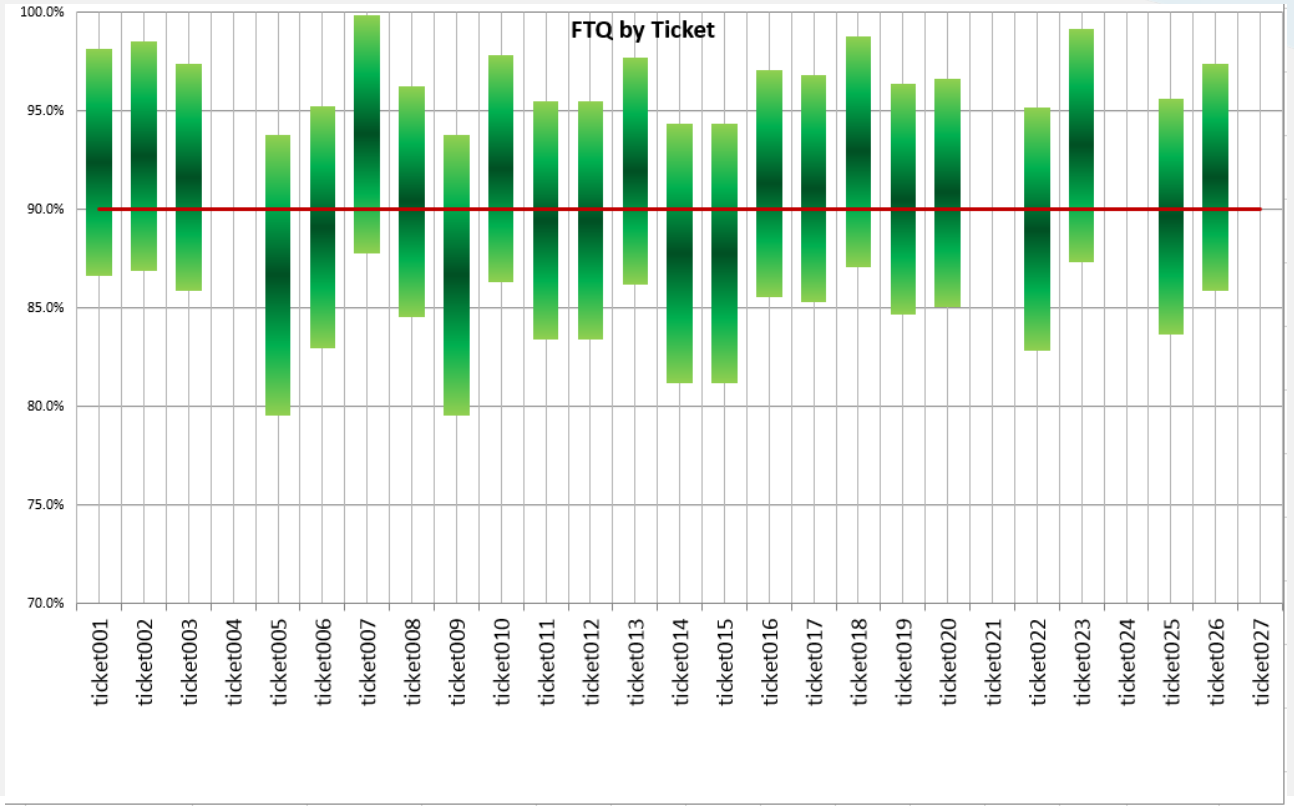


Upper Limit	333 hrs
Most Probable	206 hrs
Lower Limit	80 hrs

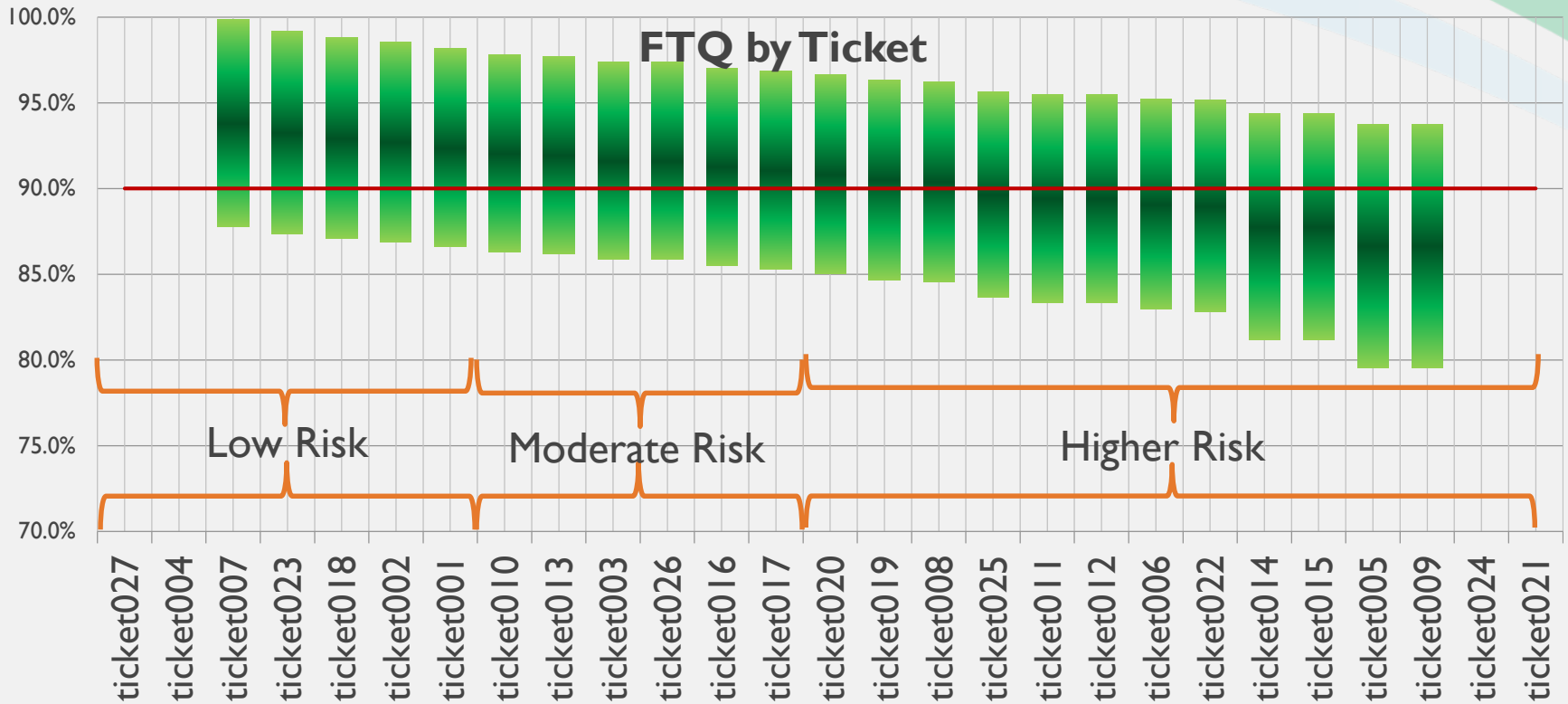


Upper Limit	162
Most Probable	101
Lower Limit	40

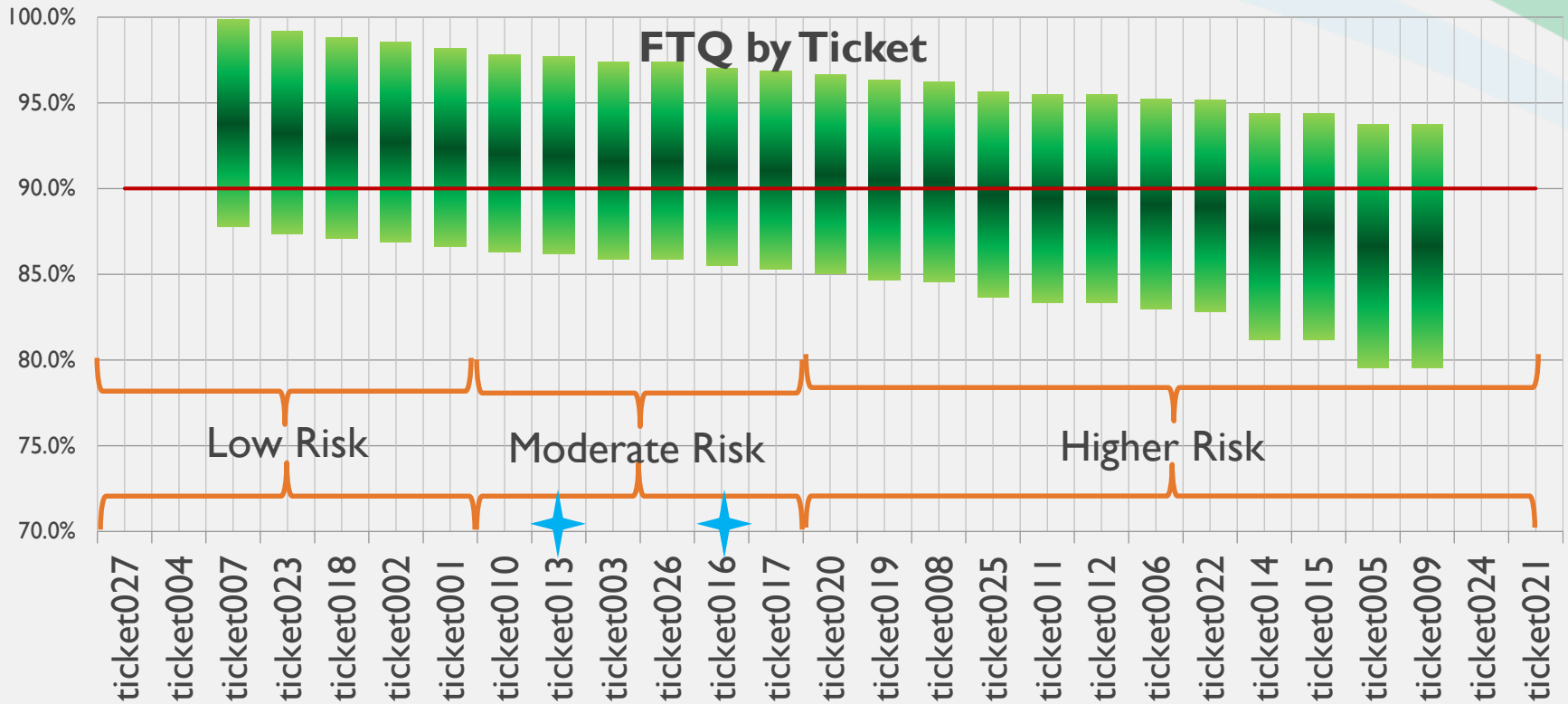
IMPROVEMENT
FTQ BY TASK



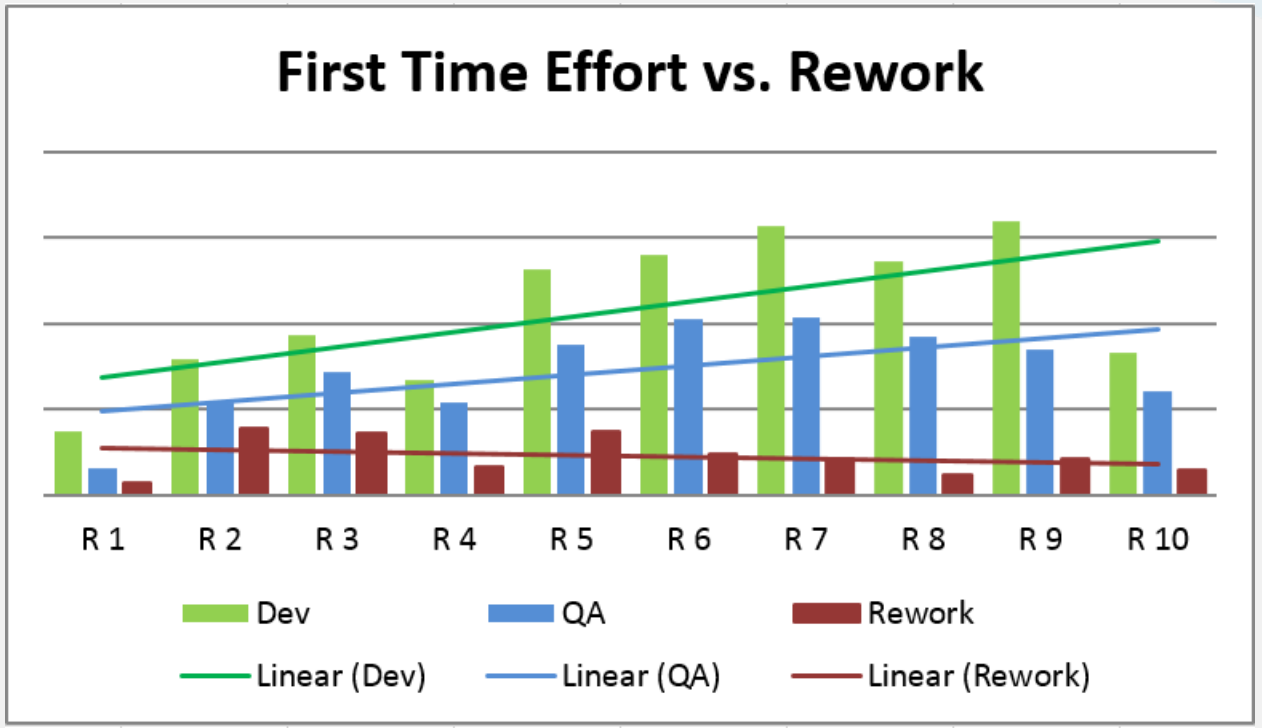
COMPOSING THE PROCESS BASED ON FTQ BY TASK



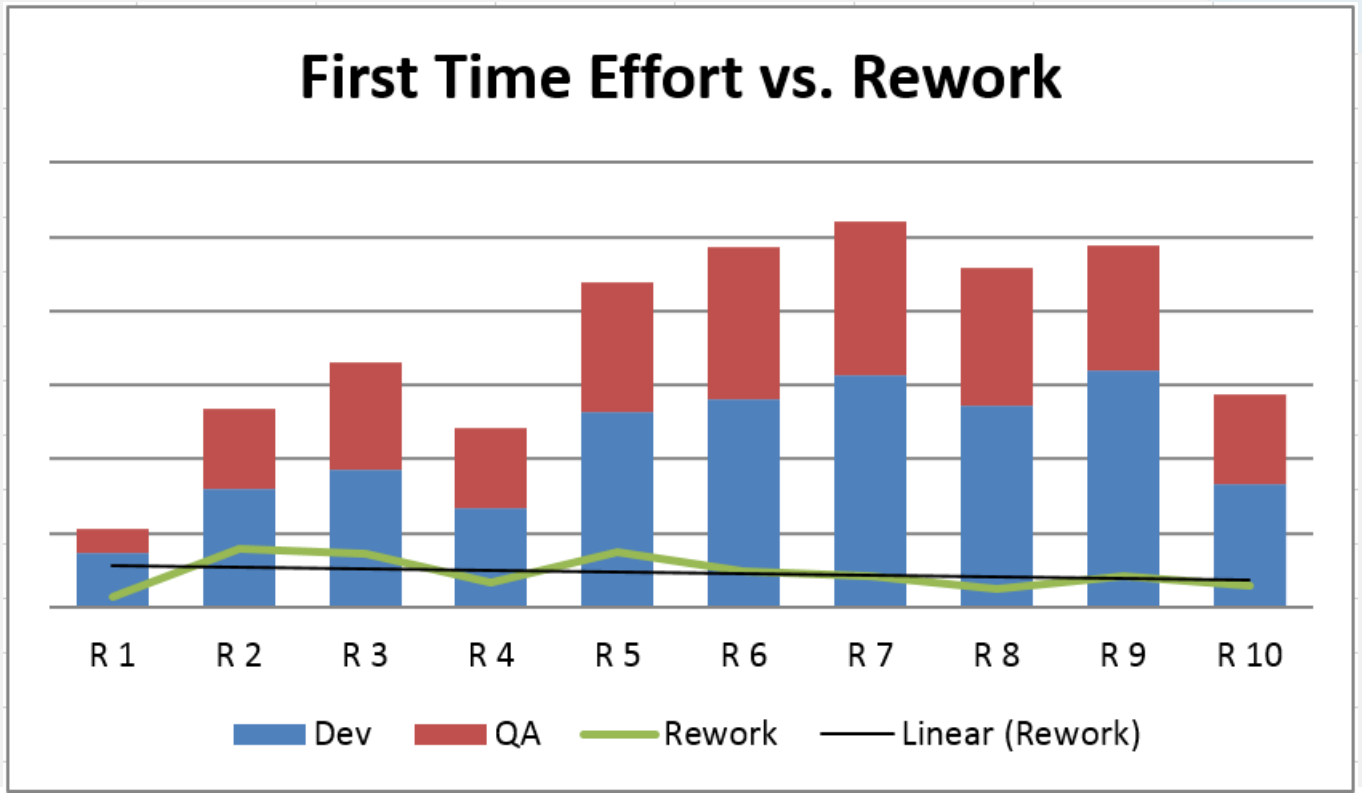
COMPOSING THE PROCESS BASED ON FTQ BY TASK



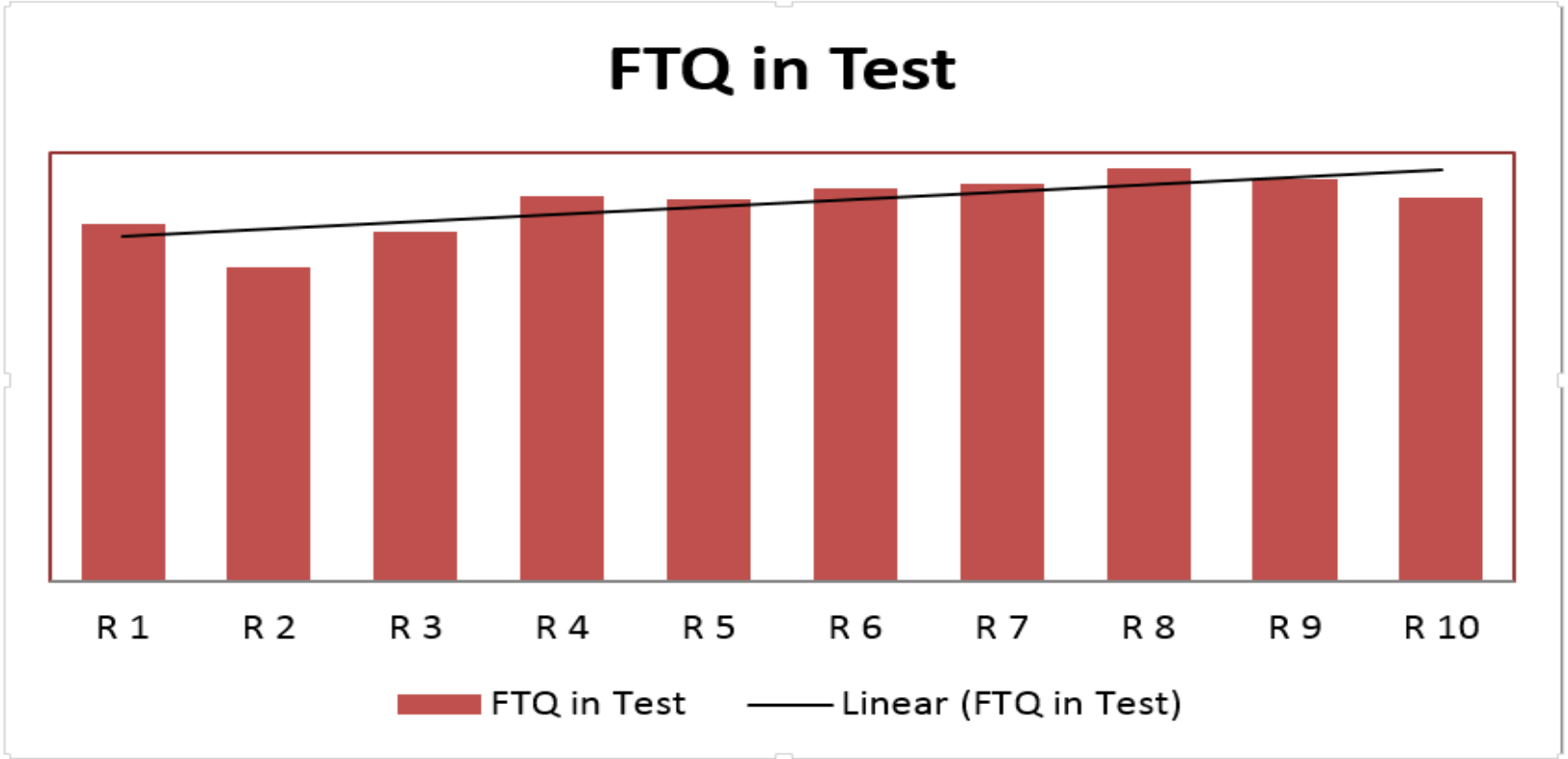
RESULTS OF USING THE NGD FIRST TIME QUALITY MODEL



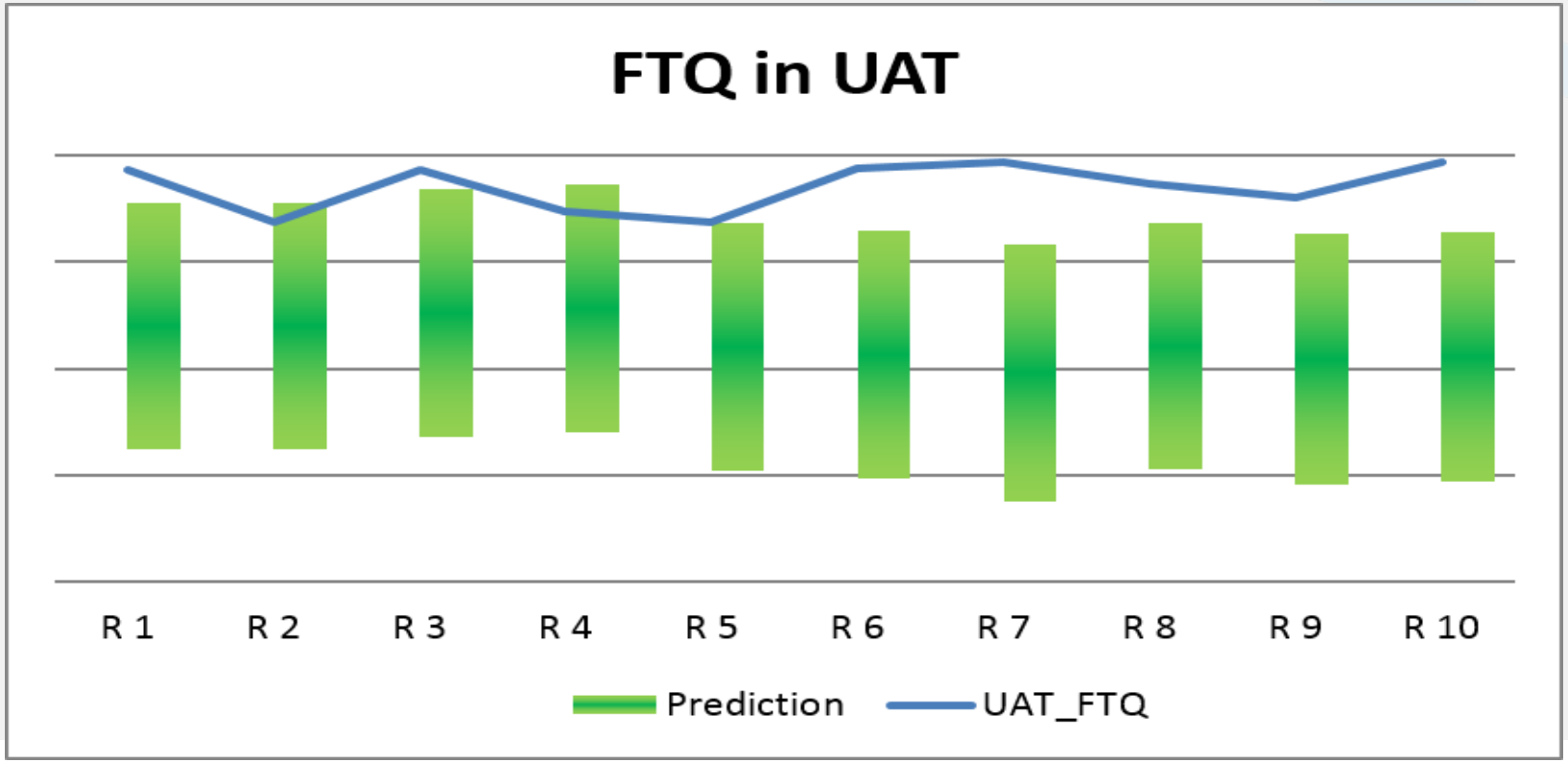
RESULTS OF USING THE NGD FIRST TIME QUALITY MODEL



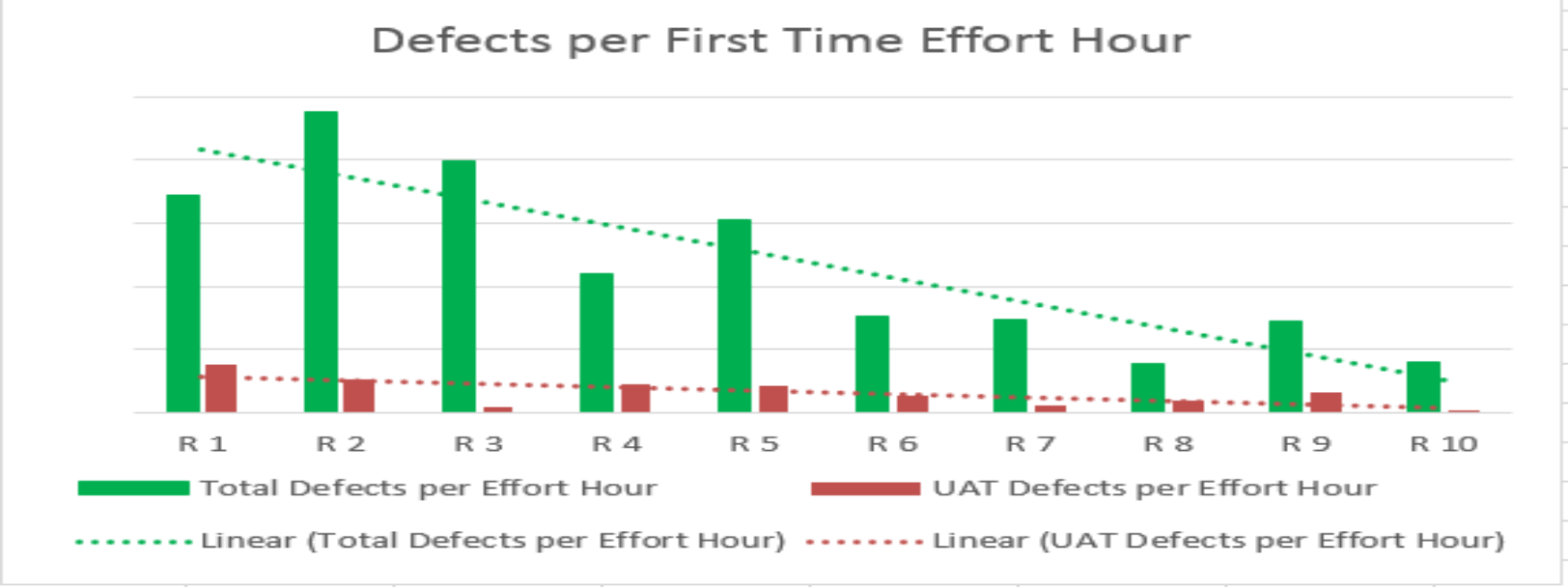
RESULTS OF USING THE NGD FIRST TIME QUALITY MODEL



RESULTS OF USING THE NGD FIRST TIME QUALITY MODEL



RESULTS OF USING THE NGD FIRST TIME QUALITY MODEL



LESSONS LEARNED

- A repeatable reproducible process for collecting input data is vital to produce useable inputs for analysis.
- Data relationships that seem intuitive must be validated statistically in order to produce useable predictive models.
- Continuous improvement to working Predictive Models can improve precision and usefulness.
- Causal Analysis and Resolution can help resolve problems with Predictive Models.
- A statistically based Predictive Process Performance Model can significantly enhance the probability of achieving critical Organizational Business Objectives.

LESSONS LEARNED
(CONTINUED)

- Application of Statistical Process Control (SPC) in the software environment, where each program has unique characteristics, is quite different than in a manufacturing environment, where large volumes of identical products are produced. Smaller sample sizes can have considerable impact on prediction precision..
- Many factors – some measureable and some not – impact software quality. In addition, there may be interactions between those factors. In order to best manage the number of possible factors and interactions, NGD has elected to work with and understand the effects of one major factor at a time.
- Given the number of factors and smaller sample size, we may need to accept that measures of the strength of relationships such as correlation coefficient (r) may not be as strong as in situations with larger sample sizes.

NEXT STEPS

- Refresh the Predictive Model with more current historical data
- Evaluate additional factors for potential inclusion as inputs to the Predictive Model.
- Review the estimating approach for potential improvement, which would narrow prediction ranges.
- Explore opportunities for additional Predictive Process Improvement Models

QUESTIONS?