STAT T&E COE-Report-09-2013

Reliability Test Planning for Mean Time Between Failures

Best Practice

Authored by: Jennifer Kensler, PhD STAT T&E COE

March 21, 2014

The goal of the STAT T&E COE is to assist in developing rigorous, defensible test strategies to more effectively quantify and characterize system performance and provide information that reduces risk. This and other COE products are available at www.AFIT.edu/STAT.

STAT T&E Center of Excellence
2950 Hobson Way – Wright-Patterson AFB, OH 45433

Table of Contents

Executive Summary

The “three times rule” remains a popular rule of thumb for planning reliability tests for the mean time between failures (MTBF). This rule sets the number of allowable failures during test at one and establishes the total test time as three times the MTBF to be demonstrated at the 80% confidence level. While simple to implement, this approach fails to consider power leading to underpowered tests that fail acceptable systems. This best practice investigates the basis, assumptions, and limitations of the three times rule and suggests an alternative approach that considers both confidence and power to determine the total test time and number of allowable failures for a test plan.

Keywords: Mean Time Between Failures, Constant Failure Rate, Confidence, Power

Introduction

A Motivating Example

Department of Defense (DoD) acquisition programs typically state reliability requirements as an MTBF. The notional Snapdragon program is developing an unmanned aerial vehicle (UAV) with an MTBF threshold requirement of 100 hours and an objective requirement of 200 hours. The Snapdragon program office uses the three times rule to set up a test to demonstrate the threshold requirement at the 80% confidence level. The three times rule means testing for three times the required MTBF and allowing only one failure. In this case, the test plan calls for 300 hours of testing with one allowable failure. If the testers observe zero or one failures during test, they will conclude that Snapdragon exceeds the threshold requirement. On the other hand, if two or more failures occur during test, they will conclude that Snapdragon does not exceed the threshold requirement. This best practice explores how this test plan implements the three times rule, the assumptions and results of the three times rule, and suggests an alternative approach.

Background

Statistical Hypothesis Testing

Statistical hypothesis testing provides a vehicle to determine whether the true, but unknown, MTBF of a system under test exceeds some value, say $MTBF_0$. In the language of hypothesis testing the null and alternative hypotheses are $H_0: MTBF \leq MTBF_0$ vs. $H_A: MTBF > MTBF_0$. For the Snapdragon program example, the null and alternative hypotheses for testing the threshold requirement are $H_0: MTBF \leq 100$ and $H_A: MTBF > 100$ respectively. The Snapdragon program can draw incorrect conclusions from this hypothesis test in two ways. First, the tester can conclude that the MTBF exceeds 100 when it does not (i.e. rejecting the null hypothesis when the null hypothesis is true). This type of mistake is called a type I error and the probability of a type I error is denoted α. Alternatively, a type II error can occur if the tester fails to reject the null hypothesis when the alternative hypothesis is true. The probability of a type II error is denoted by β. For the Snapdragon program a type II error means concluding the MTBF does not exceed 100 hours when it in fact does exceed 100 hours. Figure 1 summarizes the possible outcomes from a hypothesis test.

Figure 1

		Decision	
		Fail to Reject H_0	**Reject H_0**
Truth	**H_0 is True**	Correct (Confidence, $1-\alpha$)	Type I Error (Consumer’s Risk, α)
	H_A is True	Type II Error (Producer’s Risk, β)	Correct (Power, $1-\beta$)

The Snapdragon program set the confidence level at 80% (i.e. $\alpha = 0.20$), meaning that if the true MTBF is equal to 100 hours, the test will fail the system 80% of the time. This fact gives the program office *confidence* that if they conclude the MTBF exceeds 100 hours that it indeed exceeds 100 hours. However, the three times rule does not allow the tester to control the power (complement of the type II error rate) of the test. Power in this context means passing the system when the MTBF really is greater than 100. The appropriate power depends on the problem at hand, but should usually be at least 80%. This best practice assumes readers are familiar with the basics principles of statistical hypothesis testing. For those unfamiliar with hypothesis testing Kensler (2014) provides an introduction.

Assumptions

Major assumptions behind the three times rule (and the method proposed in this best practice) are that the failure times are independent and follow an exponential distribution. The exponential distribution is appropriate for modeling failures with a constant failure rate, meaning there is no infant mortality (i.e. early failure), wearout, or reliability growth! The probability density function (pdf) for the exponential distribution is given by

$$f(t) = \frac{1}{MTBF} e^{-\frac{t}{MTBF}} \qquad t \geq 0, MTBF > 0 \tag{1}$$

where t is the time between failures and $MTBF$ is the mean time between failures. While many systems experience a constant failure rate over their useful life, there are also many cases where modeling the failure times with an exponential distribution is not appropriate.

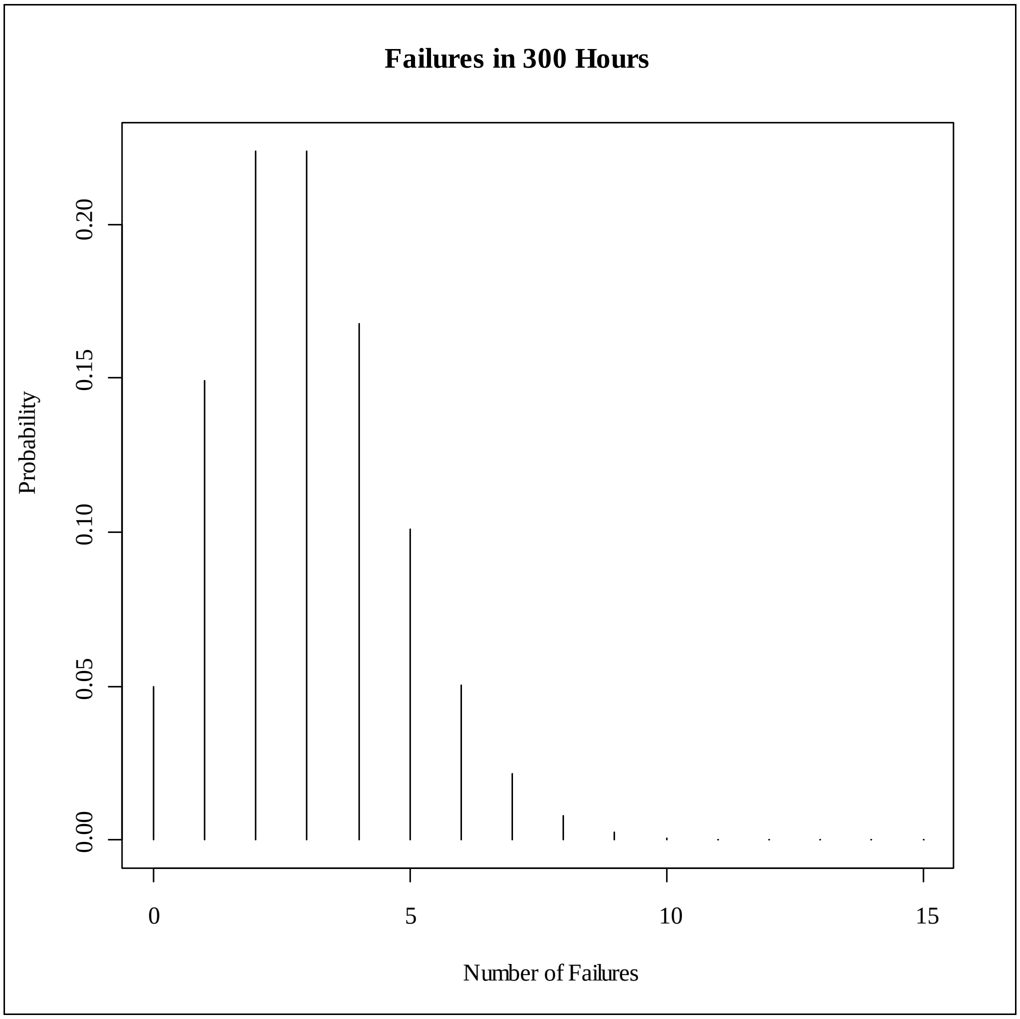
If the time between failures follows an exponential distribution, then the number of failures, R, occurring up to time T follows a Poisson distribution with probability mass function (pmf)

$$P(R = r|T) = \frac{e^{-\frac{T}{MTBF}} \left(\frac{T}{MTBF}\right)^{r}}{r!} \tag{2}$$

and $P(R = r|T)$ is the probability of exactly r failures in time T.

Figure 2 shows the probability distribution for the number of failures in 300 hours when $MTBF = 100$. Note that there is an 80% probability of observing at least two failures in 300 hours on test!

Figure 2

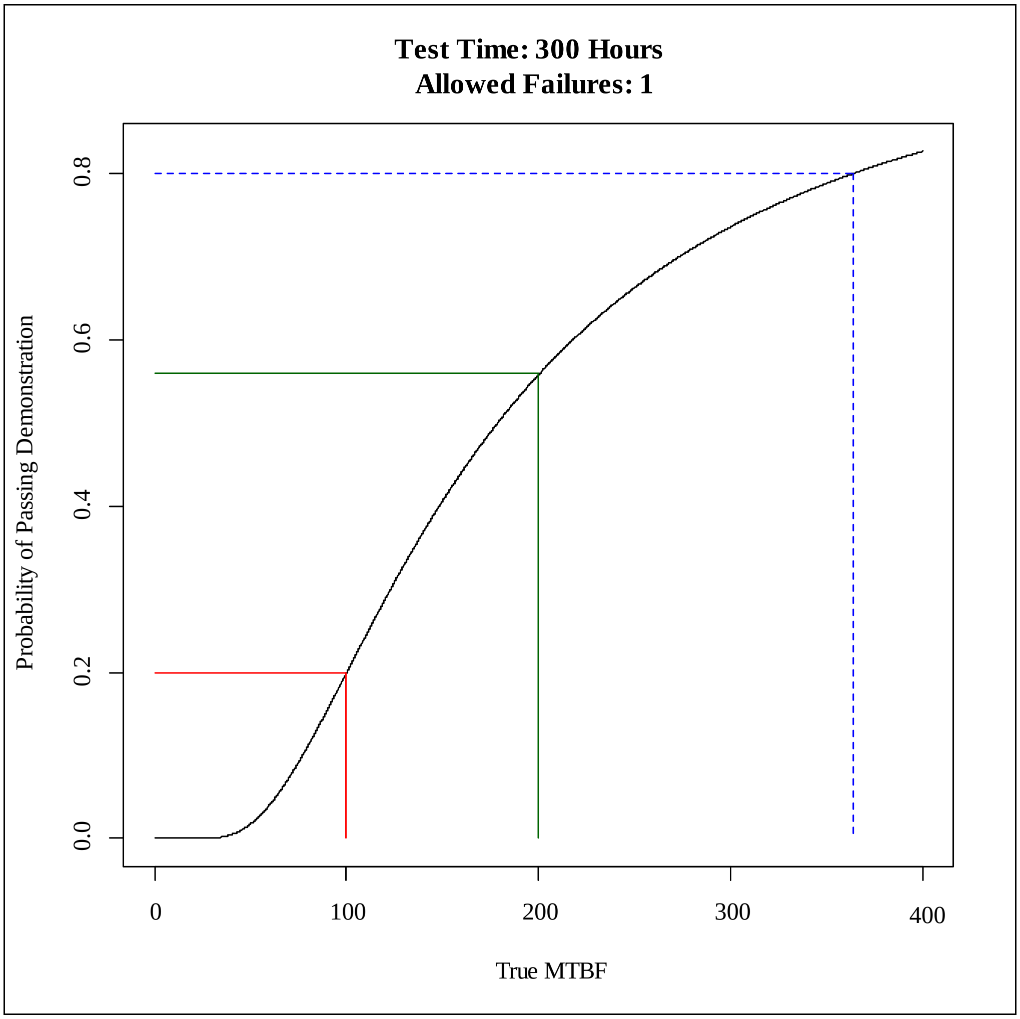


The Three Times Rule

The National Institute of Standards and Technology (NIST) Engineering Statistics Handbook (NIST/SEMATECH, 2012) provides a table to determine test length for a given confidence level and allowable number of failures. The Handbook notes that as the number of allowed failures increases the test length also increases and recommends using the largest number of allowed failures that still gives an acceptable test length because “a longer test allowing more failures has the desirable feature of making it less likely a good piece of equipment will be rejected because of random ‘bad luck’ during the test period.” This statement alludes to the low power of some short test plans, but the Test Length Guide does not formally incorporate power into the test plan. The Test Length Guide shows that for the standard DoD confidence level of 80% and one allowable failure the appropriate test time is 2.99 (approximately three) times the MTBF to be demonstrated. Note that if zero failures occur during test it is not possible to calculate a point estimate for MTBF, since the point estimate is the total test time divided by the number of failures and dividing by zero is undefined. The three times rule produces a small test at the 80% confidence level, but leaves power uncontrolled.

Figure 3 depicts the operating-characteristic (OC) curve based on the three times rule for the Snapdragon program example. This curve plots the probability of passing the demonstration versus the true, but unknown, MTBF. The figure shows that if the true MTBF is equal to the threshold requirement of 100 hours there is only a 20% chance of passing the demonstration (i.e. observing zero or one failures). If Snapdragon meets its objective requirement of 200 MTBF, then it has about a 56% chance of passing the demonstration. This test is good at failing bad systems, but it also fails good systems! Note that under the three times rule the true MTBF must be approximately 3.6 times the requirement, in this case 364 hours, in order to have an 80% chance of passing the demonstration.

Figure 3



A limitation of the three times rule is that it only considers the confidence level and fails to consider power. In this context confidence is the ability to fail a system not exceeding the threshold requirement; whereas power is the ability to pass a system exceeding the threshold requirement. Using the three times rule fails to consider power, which can result in unacceptable levels of power and literally set the test up for failure.

Method

Consideration of Power

An alternative to the three times rule is to create a test plan that balances both confidence and power. Let $MTBF_A$ be the mean time between failures required to have a high probability of passing the demonstration. In other words, $MTBF_A = MTBF_0 + \delta$, where $\delta > 0$ is the difference we want to be able to detect.

A test plan with a $100(1-\alpha)\%$ confidence level and power of **at least** $1-\beta$ has r allowable failures where r is the smallest nonnegative integer such that

$$\frac{\chi^2_{\beta,2(r+1)}}{\chi^2_{1-\alpha,2(r+1)}} \geq \frac{MTBF_0}{MTBF_A} \tag{3}$$

and total test time

$$T = \frac{1}{2} MTBF_0 \chi^2_{1-\alpha,2(r+1)} \tag{4}$$

where $\chi^2_{p,k}$ is the pth percentile of the chi-squared distribution with k degrees of freedom. Appendix A contains the derivation of these equations. A STAT COE App is available to calculate the number of allowable failures and total test time based on inputted values of $MTBF_0$, $MTBF_A$, α, and β.

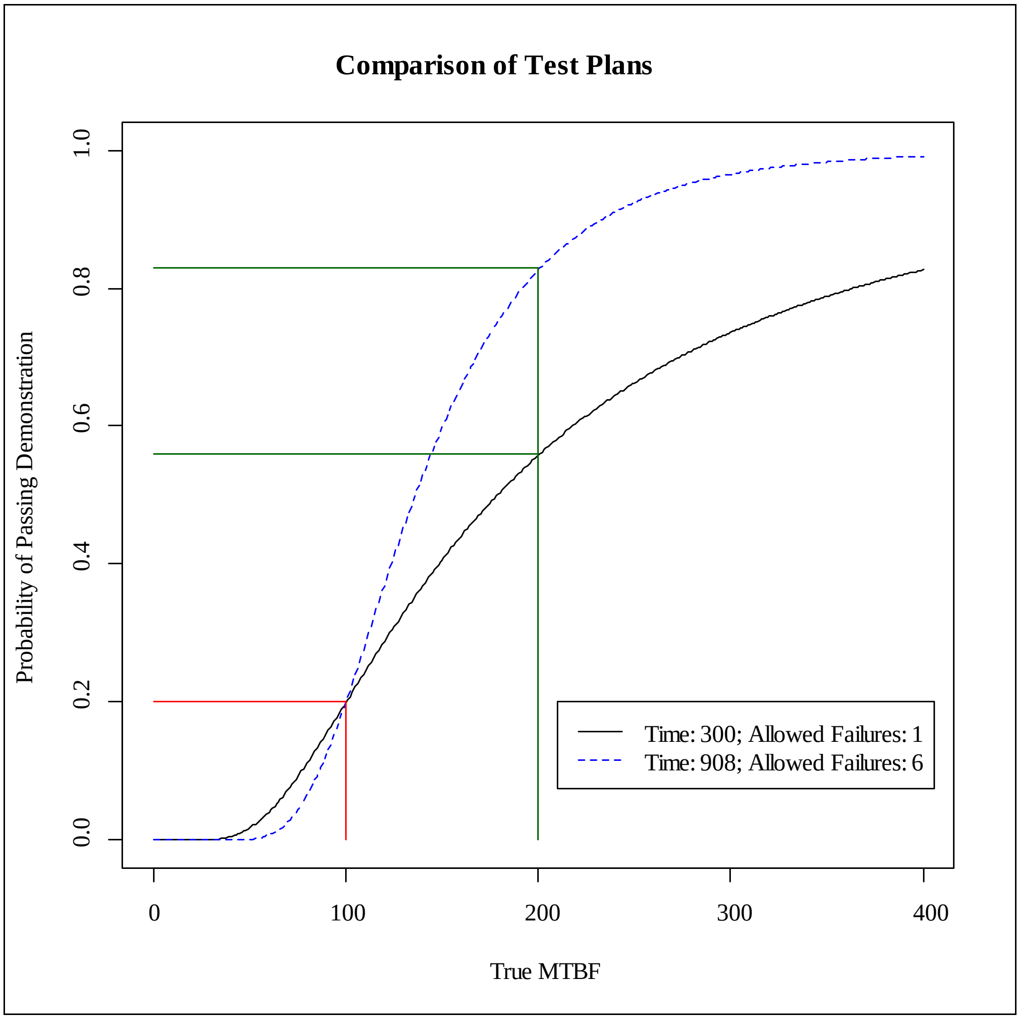
Snapdragon Program Example

Recall the Snapdragon UAV program with a threshold requirement of 100 hours MTBF and an objective requirement of 200 hours MTBF. The program office wants to demonstrate that the threshold requirement has been met at the 80% confidence level, which implies $MTBF_0 = 100$ and $\alpha = 0.20$. In order to incorporate power into the test plan we will use the objective requirement as the MTBF under the alternative hypothesis, $MTBF_A = 200$, and set $\beta = 0.20$. Thus, if the MTBF is truly 200 hours we want a test that has an 80% chance of concluding that the MTBF is greater than 100. In general $MTBF_0$ and $MTBF_A$ are not necessarily the threshold and objective requirements. The tester must carefully consider what represents unacceptable and desirable MTBFs. The value of $MTBF_0$ should reflect a system that is unacceptable and that the tester wants a high probability of failing, since the tester wants to evaluate whether the MTBF ***exceeds*** $MTBF_0$. On the other hand, $MTBF_A$ should represent a level of performance that the tester wants a high probability of passing. The smallest integer r that satisfies

Equation (3) is six; therefore, the test plan will allow six failures. Substituting r into Equation (4) produces a total test time of 907.5 hours. The actual type II error rate β is 0.174, so the power is 0.826. The Appendix includes resources for implementing this method. Appendix B shows the input and output of a STAT COE applet that creates this test plan and Appendix C provides code for creating this test plan using R.

Figure 4 compares the OC Curve for this solution with the OC Curve for the three times rule solution.

Figure 4



Although the three times rule may appear appealing in terms of test time, there is a hidden cost. Certainly testing for only 300 hours will be cheaper than testing for 908 hours. However, in order to have an 80% chance of demonstrating the requirement under the three times rule, the true MTBF would need to be approximately 364 hours. While the advantages of designing reliability into a system are numerous, this 364 hour MTBF is well above the stated threshold and objective values. In order to save money in test, the three times rule requires the system to be built to a much higher standard! The cost of designing the system to this higher standard may be more expensive than testing for 908 hours.

Conclusions

This best practice highlights the importance of considering both power and confidence in test design. Without sufficient power, it is unlikely that the system will pass the demonstration unless it is designed well above the stated requirement. In order to have a reasonable chance of demonstrating the nominal MTBF at the 80% confidence level, the three times rule requires the system to possess an MTBF approximately 3.6 times the nominal value. The cost of designing higher reliability into the system will almost certainly be more than the cost of increasing test time. This best practice illustrates a method to determine the test time and number of allowable failures for a specified confidence level and power. For ease of implementation R code is provided in the appendix and an Excel App has been created to accompany this best practice.

References

Ebeling, C. E. (2010). *An Introduction to Reliability and Maintainability Engineering* (2nd ed.). Long Grove, Illinois: Waveland Press.

Kensler, J. (2014, January 16). *The Logic of Statistical Hypothesis Testing.* Retrieved from Air Force Institute of Technology: http://www.afit.edu/STAT

NIST/SEMATECH. (2012, April). *Exponential life distribution (or HPP model) tests.* Retrieved June 20, 2013, from e-Handbook of Statistical Methods: http://www.itl.nist.gov/div898/handbook/apr/section3/apr311.htm

Appendix

Appendix A: Derivation of Test Time and Number of Allowable Failures

A test plan with a $100(1-\alpha)\%$ confidence level and power $1-\beta$ satisfies the equations

$$\alpha = P\left(R \leq r \middle| \frac{T}{MTBF_0}\right) \tag{5}$$

and

$$1-\beta = P\left(R \leq r \middle| \frac{T}{MTBF_A}\right). \tag{6}$$

Rearranging Equation (5) gives

$$\begin{aligned} \alpha &= P\left(R \leq r \middle| \frac{T}{MTBF_0}\right) \\ &= F_{Poisson}\left(r \middle| \frac{T}{MTBF_0}\right) \\ &= 1 - F_{\chi^2}\left(\frac{2T}{MTBF_0}, 2(r+1)\right) \\ 1-\alpha &= F_{\chi^2}\left(\frac{2T}{MTBF_0}, 2(r+1)\right) \\ \chi^2_{1-\alpha,2(r+1)} &= \frac{2T}{MTBF_0} \\ T &= \frac{1}{2} MTBF_0 \chi^2_{1-\alpha,2(r+1)}, \end{aligned} \tag{4}$$

where $F_{Poisson}$ is the cumulative distribution function (cdf) of the Poisson distribution and F_{χ^2} is the cdf of the chi-squared distribution. Working with Equation (6) produces

$$\begin{aligned} \text{Power} &= P\left(R \leq r \middle| \frac{T}{MTBF_A}\right) \\ &= F_{Poisson}\left(r, \frac{T}{MTBF_A}\right) \\ &= 1 - F_{\chi^2}\left(\frac{2T}{MTBF_A}, 2(r+1)\right). \end{aligned}$$

Substituting for T gives

$$\text{Power} = 1 - F_{\chi^2}\left(\frac{MTBF_0}{MTBF_A} \chi^2_{1-\alpha,2(r+1)}, 2(r+1)\right).$$

In order to ensure power of at least $1-\beta$, find the smallest integer r such that

$$1-\beta \le 1-F_{\chi^2}\left(\frac{MTBF_0}{MTBF_A}\chi^2_{1-\alpha,2(r+1)},2(r+1)\right)$$

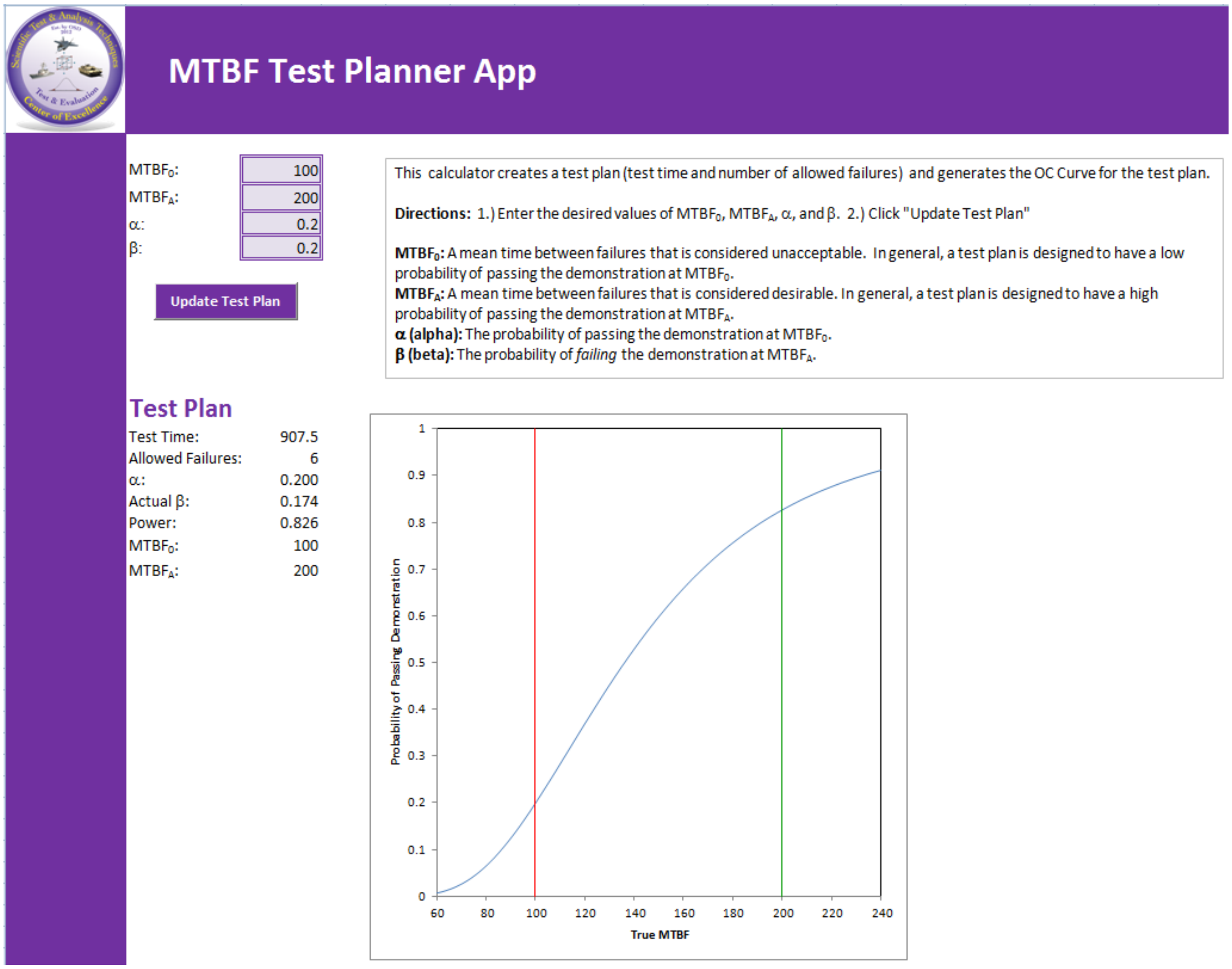
$$\beta \ge F_{\chi^2}\left(\frac{MTBF_0}{MTBF_A}\chi^2_{1-\alpha,2(r+1)},2(r+1)\right)$$

$$\chi^2_{\beta,2(r+1)} \ge \frac{MTBF_0}{MTBF_A}\chi^2_{1-\alpha,2(r+1)}$$

$$\frac{\chi^2_{\beta,2(r+1)}}{\chi^2_{1-\alpha,2(r+1)}} \ge \frac{MTBF_0}{MTBF_A}. \qquad (3)$$

Substitute the value of r obtained from Equation (3) into Equation (4) to determine the test time.

Appendix B: STAT COE Applet



MTBF Test Planner App

MTBF$_0$:	100
MTBF$_A$:	200
α:	0.2
β:	0.2

This calculator creates a test plan (test time and number of allowed failures) and generates the OC Curve for the test plan.

Directions: 1.) Enter the desired values of MTBF$_0$, MTBF$_A$, α, and β. 2.) Click "Update Test Plan"

MTBF$_0$: A mean time between failures that is considered unacceptable. In general, a test plan is designed to have a low probability of passing the demonstration at MTBF$_0$.
MTBF$_A$: A mean time between failures that is considered desirable. In general, a test plan is designed to have a high probability of passing the demonstration at MTBF$_A$.
α (alpha): The probability of passing the demonstration at MTBF$_0$.
β (beta): The probability of *failing* the demonstration at MTBF$_A$.

Test Plan

Test Time:	907.5
Allowed Failures:	6
α:	0.200
Actual β:	0.174
Power:	0.826
MTBF$_0$:	100
MTBF$_A$:	200

Appendix C: R Code

```
> Plan<-function(MTBF0,MTBFA,alpha,beta){
+ r<-0
+ while(qchisq(beta,2*(r+1))/qchisq(1-alpha,2*(r+1))<MTBF0/MTBFA) r<-r+1
+ T<-MTBF0*qchisq(1-alpha,2*(r+1))/2
+ Beta<-pchisq(MTBF0/MTBFA*qchisq(1-alpha,2*(r+1)),2*(r+1))
+ list(T=T,r=r,Beta=Beta)
+ }
> Plan(100,200,0.2,0.2)
$T
[1] 907.5385

$r
[1] 6

$Beta
[1] 0.1738087
```