Anritsu

Discover What's Possible™

White Paper

Discover how to reduce development costs & time to market for mobile devices

TABLE OF CONTENTS

Introduction

Developers of mobile wireless chipsets and devices are facing increasingly difficult challenges to deliver quality, leading edge products on time and in budget. Today's smartphone will be yesterday's in 6-12 months so time-to-market is critical to secure sales, gain market share and command a premium price. Delayed product launches cannot be tolerated. Devices now encompass different cellular radio standards alongside short range technologies such as Wifi and Bluetooth. The interactions and interworking between these technologies have led to an explosion in technical complexity. The high level of demand for development and test engineers who are expert in the latest protocols and standards has resulted in a scarcity of skilled resource. All of these challenges are faced against the backdrop of uncertain market conditions leading to shrinking budgets which further increases pressure on stretched R&D teams.

Delayed product launches cannot be tolerated

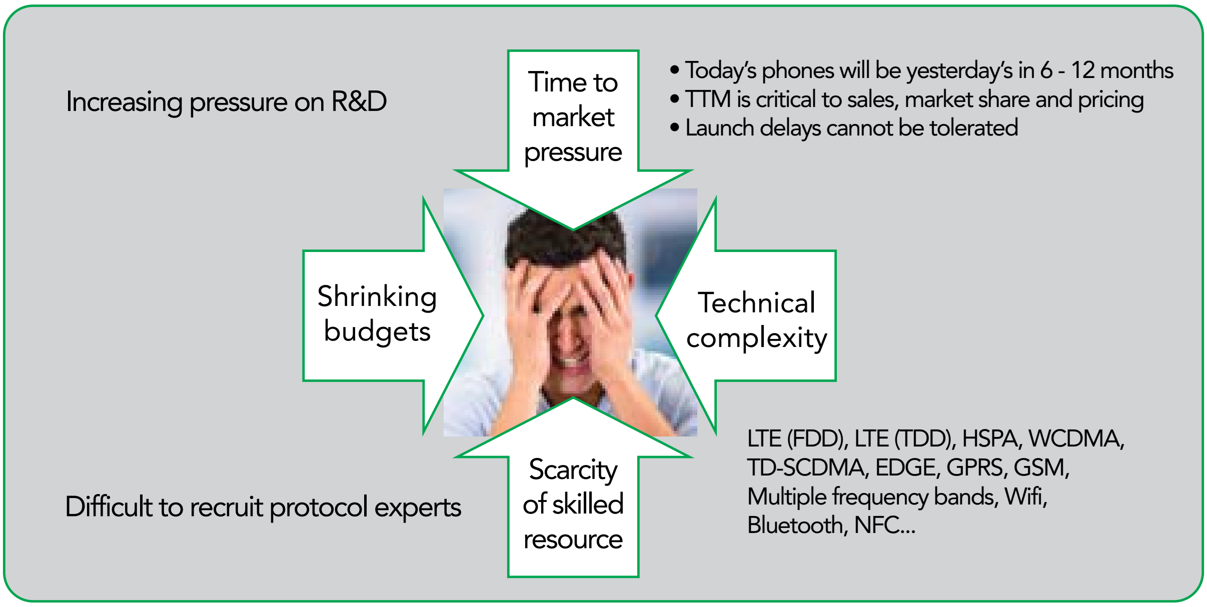

Increasing pressure on R&D
Time to market pressure
• Today's phones will be yesterday's in 6 - 12 months
• TTM is critical to sales, market share and pricing
• Launch delays cannot be tolerated
Shrinking budgets
Technical complexity
Scarcity of skilled resource
Difficult to recruit protocol experts
LTE (FDD), LTE (TDD), HSPA, WCDMA, TD-SCDMA, EDGE, GPRS, GSM, Multiple frequency bands, Wifi, Bluetooth, NFC...

Figure 1 - It is increasingly difficult to deliver on time and in budget

This paper explains why Anritsu's Rapid Test Designer (RTD) solution is the first choice of tool for the leading mobile protocol developers who successfully overcome these challenges. It describes the tangible benefits that result from the adoption of this innovative approach. It demonstrates how RTD is the signaling tester solution that delivers all of these benefits to the greatest extent – providing exceptional return on investment value.

Business Goals

Identify and Fix Problems Early

Fixing problems early is a business goal for any wireless product development organization

Fixing problems at the earliest opportunity will minimize your costs, risk and time to market. The financial benefits of fixing problems early have an order of magnitude that runs into millions of dollars. Finding and fixing a software problem after delivery is typically 100 times more expensive than finding and fixing it during the requirements and design phase*. Problems experienced by consumers in the field can be catastrophic for a brand. 6 of the top 10 most valuable global brands in 2013 are telecoms and technology companies with a combined brand value of $612bn**. Reduction in consumer confidence that results from field issues will quickly decimate your brand equity. Identifying and fixing problems early is a business goal for any mobile product development organization.

Implement Correctly – First Time Round

Correctly implementing protocols the first time round will reduce cost and improve product quality

Your products need to work in accordance with the latest industry standards to ensure a high quality user experience, interoperability with different networks and global roaming capability. Any deviation from the product or standards specifications will need to be fixed. Between 20-80% of your development effort can be spent on rework*. This extra engineering effort is frequently under-estimated in the original plans and results in a launch delay, a reduction in functionality or a reduction in product quality. Faced with an immovable launch date, working towards a goal of correctly implementing protocols the first time round will reduce your costs and improve product quality.

*The Center for Empirically-Based Software Engineering (CeBASE) eWorkshop, 2001.

** http://www.marketingweek.co.uk/Journals/2013/05/20/x/b/m/BrandZ-Top-100-2013.pdf

Commercial Signaling Testers

A commercial signaling tester will help you to meet your business goals of early fixes and correct implementation.

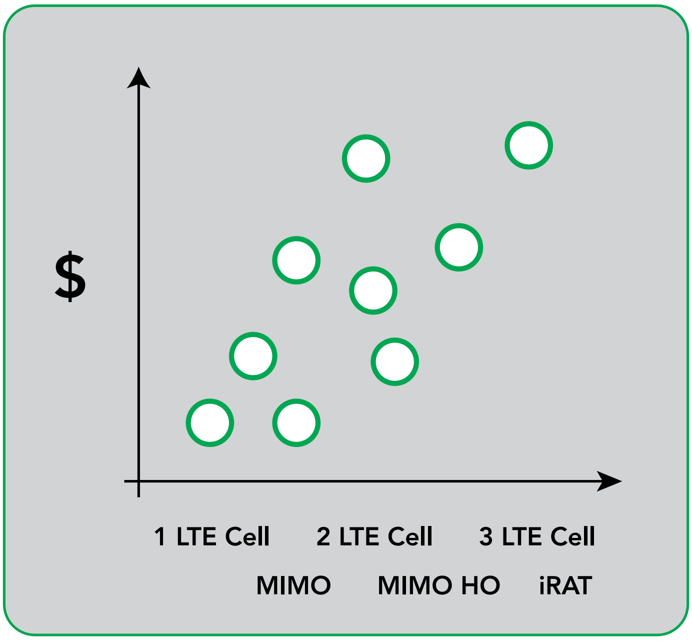
Commercially available signaling testers incorporate software tools for creating tests, running tests, and analyzing UE behavior. A properly designed signaling tester offers you different points of observation and control along with the ability to change any parameter of the simulated network. This means that you can isolate issues with the UE under test and observe behavior under a wide range of conditions – including exceptional conditions that are beyond 'normal' operation. The very best purpose-built signaling testers are offered by companies with a long-standing tradition in test and measurement with a deep understanding of the specific needs of UE developers. Significant benefits will be gained by adopting a purpose-built signaling tester as the heart of a successful UE development and testing strategy.

> *The very best purpose-built signaling testers are offered by companies with a long-standing tradition in test and measurement*

- **Reliability** for repeatable tests
- **Accuracy** for confidence
- **Adaptability** easily create test variants
- **Technology lead** test new technology early
- **Speed** fast test creation and automation
- **Focus** on core activities
- **Scalability** replicate for consistent result

There are a number of commercially available signaling testers that incorporate software tools for creating tests, running tests, and analyzing the behavior of the UE under test. How can you begin to choose between them?

The signaling tester hardware and its associated software tools are available at a range of price vs. Performance points. When you are selecting a signaling tester it is important to stay focused on the business goals and consider the extent to which each alternative on offer will maximize the benefits to your organization. You will also want to invest in a platform that has a road map to support future technology evolutions and to select a trusted supplier with whom you can develop a long term relationship.



Anritsu's Rapid Test Designer – RTD

The Rapid Test Designer (RTD), coupled with Anritsu's preeminent range of mobile signaling testers has been adopted by the world's leading device makers and network operators as the unsurpassed solution for rapidly testing and debugging wireless mobile device signaling.

The RTD software, used in conjunction with one or more Anritsu signaling units provides 3 core functions:

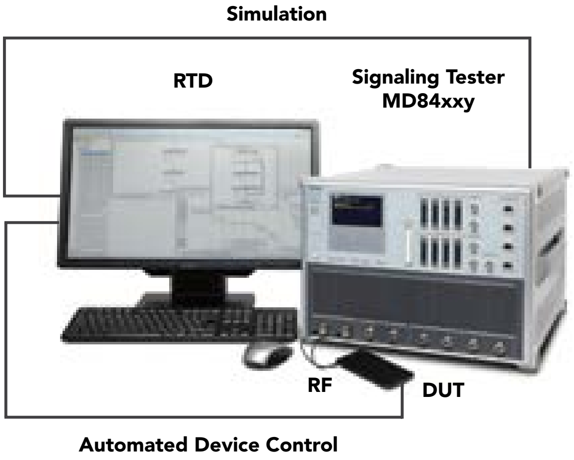
1. Custom Test Design

Quickly and easily create your own bespoke tests in a flowchart-based test design environment.

2. Mobile Network Simulation

Configure one or more base station simulators and control the messages exchanged with the Device Under Test (DUT).

Technology	Signaling Tester
LTE - Advanced, LTE	MD8430A
HSPA+, WCDMA	MD8480C
(E)GPRS, GSM	MD8480C
TD-SCDMA	MD8475A



3. Protocol Analysis

Best-in-class protocol analysis capabilities allow you to easily analyze test procedures and messages to pinpoint and fix complex protocol issues.

Why is RTD the Most Popular Choice?

Let's consider how RTD maximizes each of the business benefits arising from the use of a signaling tester.

Reliability

Anritsu's signaling testers have an unsurpassed reputation for reliability. We put every release of RTD software through a rigorous testing process. As one part of our testing process we run every reference test 30 times in succession to ensure test-retest repeatability. This increased confidence means you can rely on a test failure as indication that there is a real issue that needs investigating. You will resolve more issues earlier, thus saving time and money at a later more critical point in your project.

Accuracy

The accuracy of an RTD system exceeds the minimum level expected from a protocol tester. This is because the same Anritsu signaling testers used with RTD are also used in our market leading ME7873 RF Conformance Test System which is used for device type approval – an area which demands the most stringent of measurement accuracies.

Adaptability

RTD gives you the ultimate trade-off between ease-of-use and control. Testing of a complete stack should not require the user to understand the parameterization of the lower layers but testing a partial stack implementation (e.g. Physical layer only) will need full control over the low layer protocols. RTD achieves this by offering 2 levels of protocol procedures. 'Layer 3 procedures' enable fast and error-free test creation through automatic configuration of layers 1 and 2. 'Low level' procedures deliver fine-grained control of layers 1 and 2. Time and money is saved - not only through rapid test creation, but also because RTD is the correct tool for each job.

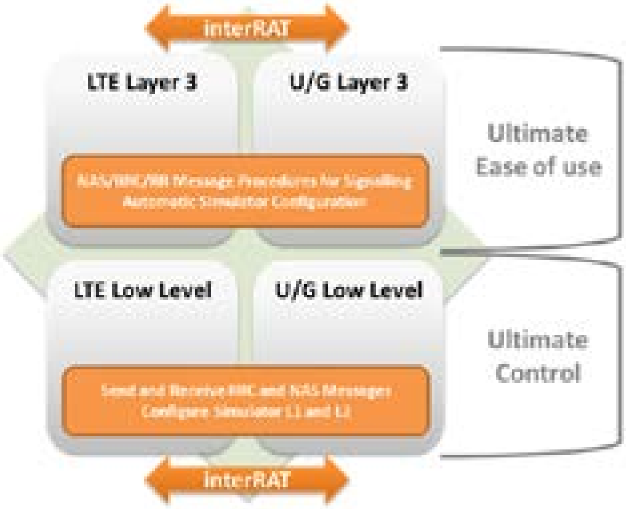

interRAT
LTE Layer 3
U/G Layer 3
NAS/RRC/RB Message Procedures for Signalling
Automatic Simulator Configuration
Ultimate Ease of use
LTE Low Level
U/G Low Level
Send and Receive RRC and NAS Messages
Configure Simulator L1 and L2
Ultimate Control
interRAT

Figure 2 - RTD gives you the ultimate trade-off between ease-of-use and control

Technology Lead

Anritsu's RTD remains ahead of every wireless technology wave. We partner with the wireless innovators and have a track record of delivering support for new technologies first. For example:

- In 2001 we supplied the world's first WCDMA signaling tester
- 10 years later we were the first to demonstrate LTE-Advanced capabilities on the leading LTE signaling tester

This means that RTD will be ready to test new features as soon as you implement them. You will not need to develop your own test harnesses to exercise your leading edge functionality.

Speed

RTD's flowchart based scripting interface speeds up your test creation, and its protocol analysis features help you to pinpoint issues fast. RTD tests are interpreted – not compiled, which means that after a test has been modified it is ready to run straight away. During early development when many debug iterations are needed this saves many hours of precious time.

RTD can automatically control a UE for unattended operation and the test sequencer can run regression suites containing thousands of tests. This means that instead of spending time running tests, you can spend your time on more valuable activities - analyzing results and fixing problems.

Mobile devices are typically developed by multiple remote teams. The self-documenting nature of RTD test scripts and the consistent procedure naming means that tests can be picked up and understood by people other than the original test creator much faster than alternatives such as complex 'C' programs.

The 3GPP protocol specifications are regularly updated and frequently RTD software updates are provided in order to track the latest standards. RTD's automatic script upgrade feature means that you no longer have to spend hours reworking all the tests in your regression suite to work according to the latest release of the standards.

Business Focus

Unlike traditional solutions RTD offers a unique flow chart based scripting interface. It has several characteristics which take the pain out of test development and execution and allow you to focus on your core business.

(1) Flowchart Based

Traditional test scripting environments are based around general purpose programming languages such as 'C'. RTD's test design interface has been designed by protocol engineers, focused on testing UE implementations.

(2) No Requirement for Programmers

RTD's procedure names match up with the industry specifications, so they are familiar to protocol engineers and defining a test flow is intuitive to non-programmers. A good understanding of the test objectives and the protocols under test will always be required, but an RTD user does not need to be an experienced software engineer or computer programmer. You can focus your expert software engineers and programmers on your core revenue-generating business activity of product development. Furthermore, it is now possible for you to hire lower cost staff from a larger pool. For example, an engineer with 3GPP protocol skills and 5 years of programming experience might expect a salary 25% greater than a similar engineer with 2 years of programming experience*. You probably spend thousands of dollars hiring a new recruit and it will typically take around 8.2 weeks to fill a technical vacancy**, but with RTD it will be easier and faster (and therefore cheaper) to recruit staff.

*Obtained by creating comparative salary reports at www.payscale.com, 9th July 2013

** TalentPuzzle, 7th July 2011 http://blog.talentpuzzle.com/online-recruitment-blog/what-is-the-average-time-to-hire/

(3) Faster to Learn = Lower Training Costs

RTD's integrated test development environment and intuitive procedure libraries are easier to learn than any other system. We don't pretend that protocol development and testing is easy. All users will require some level of training in order to become productive, but with RTD even totally novice users have created a complex test in less than 3 hours.

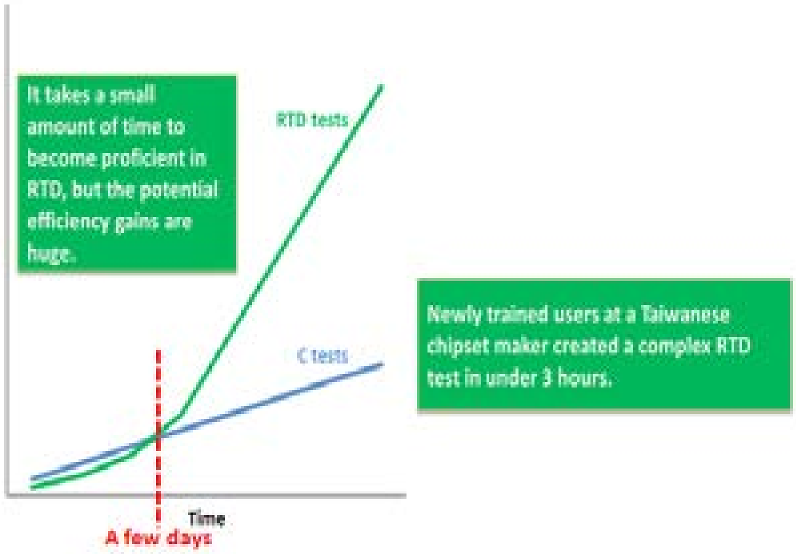


Figure 3 - RTD is fast to learn, which means lower training costs

(4) Integrated Design/Execution Environment

RTD is an integrated test design and execution environment that installs as a single software package. You do not need to supply additional software such as code editors and debuggers, and then spend time setting up (and fixing issues with) the test software environment. With other traditional systems this process has been known to consume days of precious engineering time.

(5) Test Management Features

Your teams will invest in the creation of thousands of tests during their project. This collection of tests encapsulates critical knowledge and is a significant asset to your business because it can be reused, shared among teams and extended for use on different development projects. Like any other asset it needs careful management if you are to get the most from it. RTD allows you to easily manage sharable project workspaces and archives, and you can store tests within your configuration control system.

Scalability

RTD is available as a complete test creation, execution and analysis environment. You can also buy the test design and analysis software 'standalone' so that your engineers are not constrained by limited access to a hardware platform. You save time and money by activating RTD functions from a shared network server with 'floating' licenses. This makes the software available exactly when and where you need it. RTD protocol logs can be exported to HTML format, allowing you to share them with anyone with an internet browser - no additional software is required. RTD also shares many components with our ME7834 Mobile Device Test Platform, so you can easily extend your system to perform conformance testing against the 3GPP specifications and carrier-specific acceptance test plans.

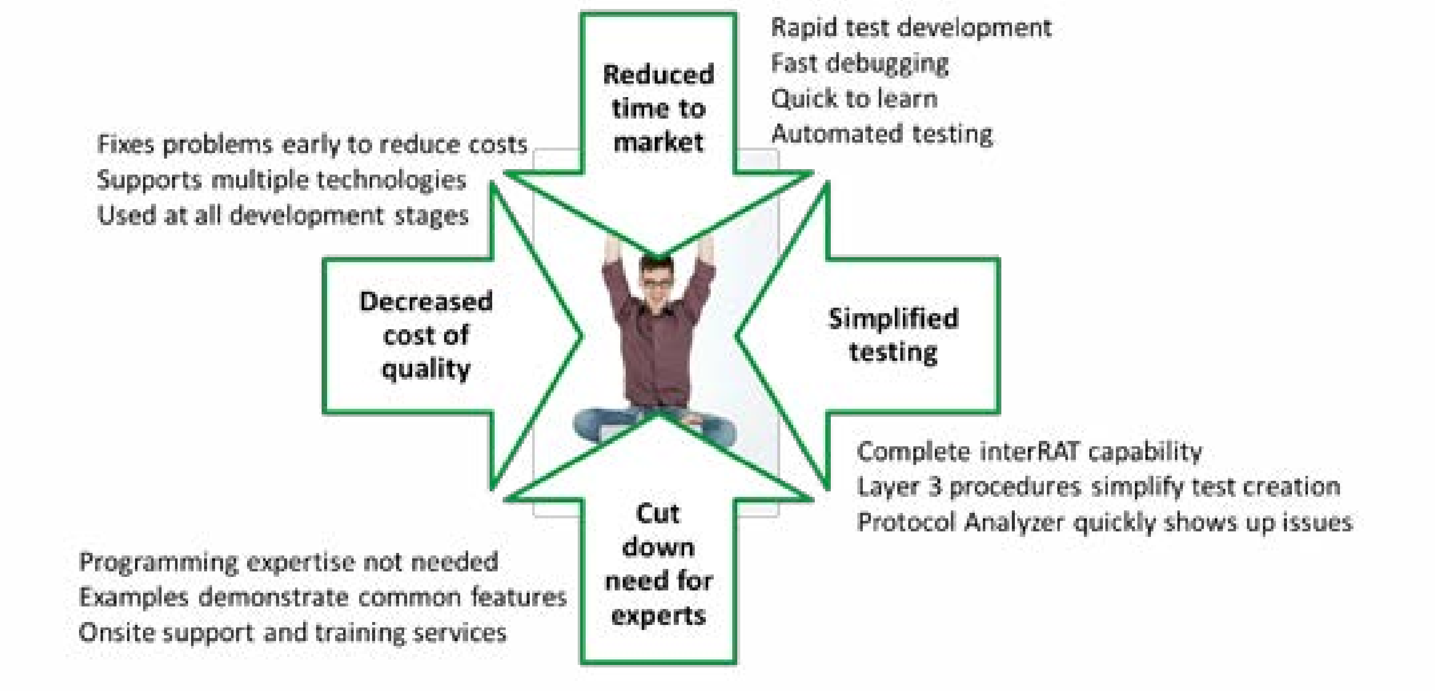

Reduced time to market
Rapid test development
Fast debugging
Quick to learn
Automated testing
Fixes problems early to reduce costs
Supports multiple technologies
Used at all development stages
Decreased cost of quality
Simplified testing
Complete interRAT capability
Layer 3 procedures simplify test creation
Protocol Analyzer quickly shows up issues
Programming expertise not needed
Examples demonstrate common features
Onsite support and training services
Cut down need for experts

Figure 4 - RTD is the ideal solution

Time to Market and Cost Savings Model

It is difficult to quantify the benefits of early problem fixes and faster time to market in hard cash terms, but one may safely assume that it is a high value measured in millions of dollars. This is why world-class organisations are performing as much testing as possible using signaling testers in their laboratories.

Fortunately it is easy to quantify the expected time to market improvements and operational cost savings that Anritsu's RTD can deliver relative to your current method. Based on feedback from our customers we have developed a model that allows you enter your own parameters and assumptions to instantly calculate the cost savings and time to market advantage that RTD will deliver to your organization.

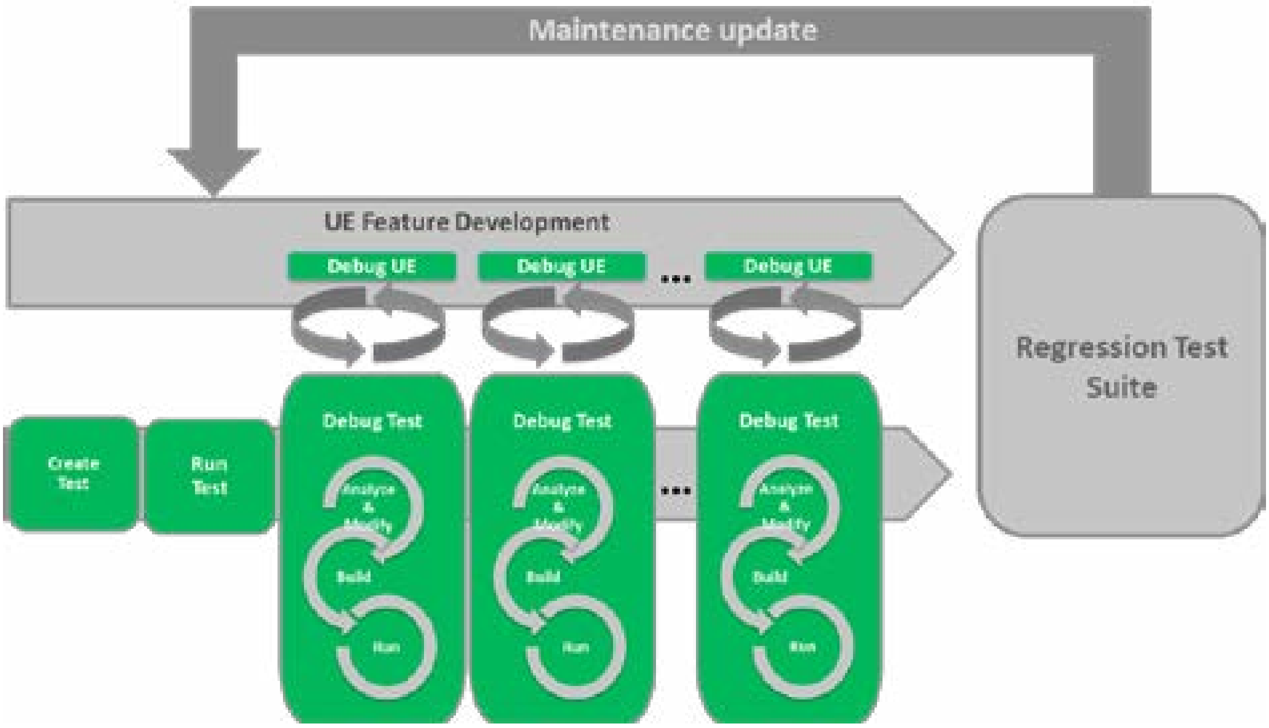
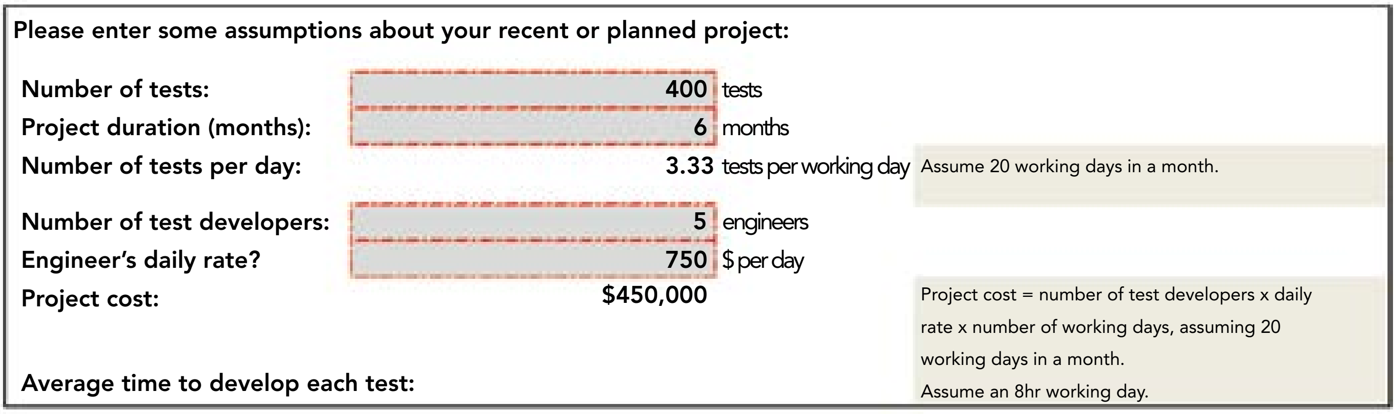


Figure 5 - RTD supporting a typical UE development process

Our model assumes a typical UE development process where a test for a protocol feature is defined and created before the feature is implemented. The test is run against the initial implementation, the results are analyzed and any required changes are made to the UE code and/or the test script. The process of test execution, analysis and modification is repeated until the feature implementation and the test script have been verified. Once verified, the test script is added into an automated regression suite that is used to verify that further UE changes have not broken existing functionality. Every so often the UE software will be updated to conform to the latest release of the 3GPP standard. To ensure that the tests in the regression suite remain valid they will need to undergo a 'maintenance update'.

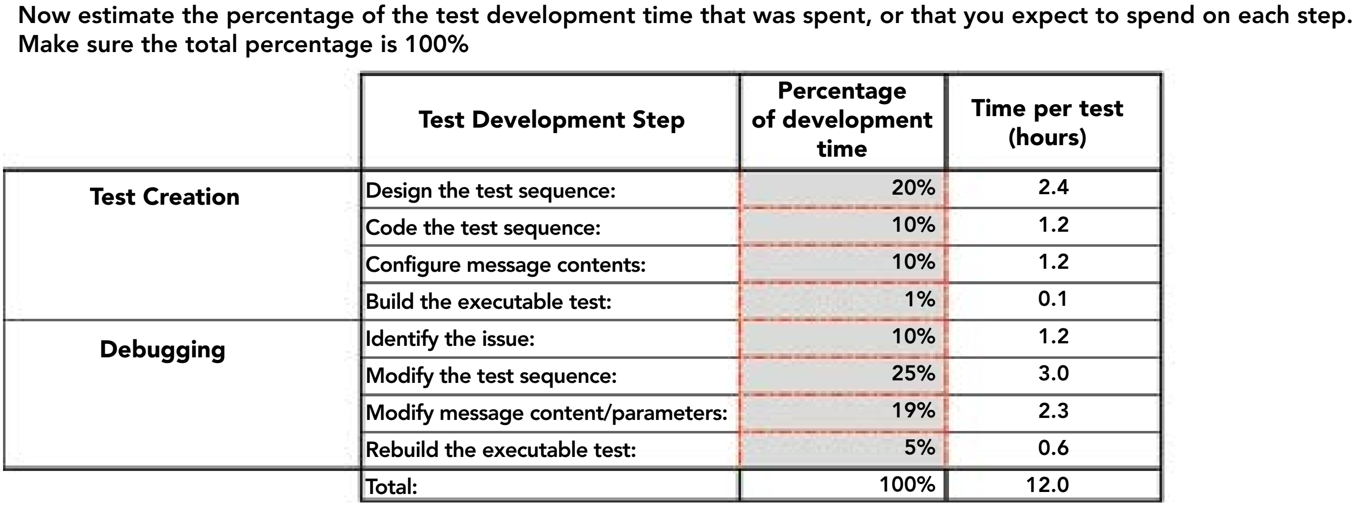
Let's walk through the model using some parameter values that we understand to be typical based on feedback from real customers:



Please enter some assumptions about your recent or planned project:			
Number of tests:	**400**	tests	
Project duration (months):	**6**	months	
Number of tests per day:	**3.33**	tests per working day	Assume 20 working days in a month.
Number of test developers:	**5**	engineers	
Engineer's daily rate?	**750**	$ per day	
Project cost:	**$450,000**		Project cost = number of test developers x daily rate x number of working days, assuming 20 working days in a month.
Average time to develop each test:			Assume an 8hr working day.

Consider your current situation. Let's say a typical project requires 400 tests to be developed within a 6 month timescale. If we assume a team of 5 engineers working an 8 hour day for 20 days of each month, then the average time to develop each test is 12 hours. The total cost of the team is $450k.

Next, we break down the test development activity into tasks and consider the proportion of the total test development time that is spent on each task. Remember that there will be several debug iterations per test, so we need to consider the total time spent on debugging tasks over all iterations.



Now estimate the percentage of the test development time that was spent, or that you expect to spend on each step. Make sure the total percentage is 100%

	Test Development Step	Percentage of development time	Time per test (hours)
Test Creation	Design the test sequence:	20%	2.4
	Code the test sequence:	10%	1.2
	Configure message contents:	10%	1.2
	Build the executable test:	1%	0.1
Debugging	Identify the issue:	10%	1.2
	Modify the test sequence:	25%	3.0
	Modify message content/parameters:	19%	2.3
	Rebuild the executable test:	5%	0.6
	Total:	100%	12.0

Now we can consider each task and look at specific RTD features that will reduce the time taken. RTD saves time in your engineer's day to day activities. There is no longer a need for separate design and code activities because the test, designed as a sequence of linked procedures will run exactly as it is! The content of protocol messages is already defined within RTD and set to usable default values. You no longer need to search through the 3GPP specifications and define your own data structures. You no longer need to search through product manuals to understand which parameters and ranges are valid for a particular function call. Furthermore, RTD allows you to easily separate the test flow from the message contents using a feature known as a 'catalog'. You can use a catalog to define the message contents once at the beginning of your project and apply the catalog to all of your tests! Customers have told us that configuring message contents is between 5 and 20 times faster with RTD than with a traditional 'C' based solution. To help with debugging, RTD allows you to monitor channels, radio bearer configurations and message sequences while the test is running. You can understand an issue even before the test has completed! RTD's flowchart-based editor then allows you to rapidly add or remove procedural blocks, or change the test flow ready for the next iteration. Identifying issues and modifying tests in RTD is between 2 and 4 times faster than with a traditional solution.

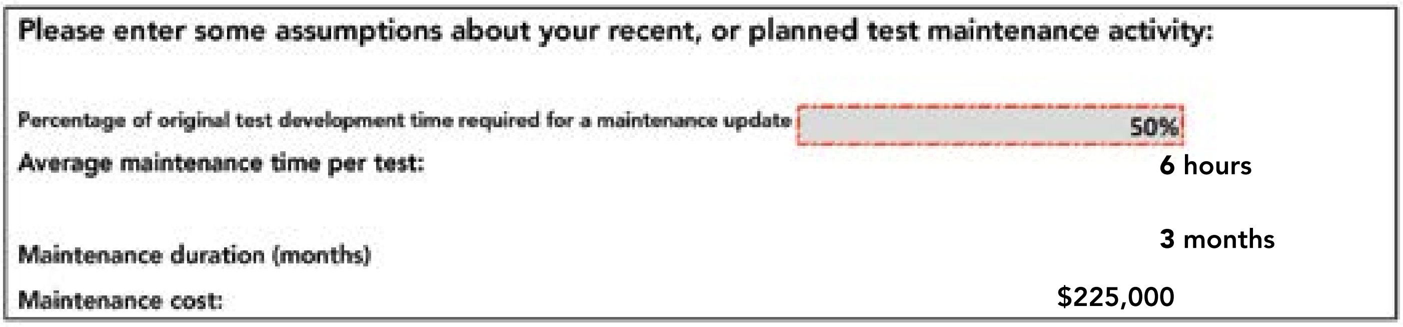
Now let's see the improvement that RTD will bring:

	Test Development Step	RTD Advantage	Expected RTD improvement factor		Time per test (hours)		Current Situation
			Min.	Max.	Min.	Max.	
Test Creation	Design the test sequence:	Test design = Test script	1	1	2.4	2.4	2.4
	Code the test sequence:	*No separate design and code steps*	N/A		0.0	0.0	1.2
	Configure message contents:	Predefined message content Parameters set to 3GPP defaults *No more searching through specs and code* Test flow separated from message contents *Define message content once, apply to all*	5	20	0.1	0.2	1.2
	Build the executable test:	RTD tests are interpreted, not complied *No build step*	N/A		0.0	0.0	0.1
Debugging	Identify the issue:	Run time monitoring *Identify issues straight away - while the test is running*	2	4	0.3	0.6	1.2
	Modify the test sequence:	Flowchart-based IDE *Add or remove procedural blocks faster*	2	4	0.8	1.5	3.0
	Modify message content/parameters:	See 'Configuring message contents' above	5	20	0.1	0.5	2.3
	Rebuild the executable test:	RTD tests are interpreted, not complied *No build step*	N/A		0.0	0.0	0.6
				Total:	3.6	5.2	12.0

If we factor these improvements into our model we calculate that using RTD **it will typically take between 3.6 and 5.2 hours to develop a test, compared with 12 hours using a traditional solution.** This is consistent with our customers' experience.

This improvement has a significant impact to your overall project. **Assuming these reduced test development times, your project will take between 1.8 and 2.6 months instead of your original 6 months. If we translate this into money, then RTD will save you between $255,150 and $314,100 in test development costs for just this single project. That's a 57% to 70% cost and time to market saving!**

Of course test development is not the only consideration. Your regression test suite is a valuable asset that must be maintained. Tests must be modified to verify a new stack design, or to verify the existing design after it has been updated to conform to a later release of the 3GPP specifications. With a traditional programming language approach, the link between the original specification and the implementation becomes more difficult to identify with every iteration. The maintenance cycles can become increasingly time-consuming until it becomes easier to completely rewrite the test suite than to update the legacy code base!

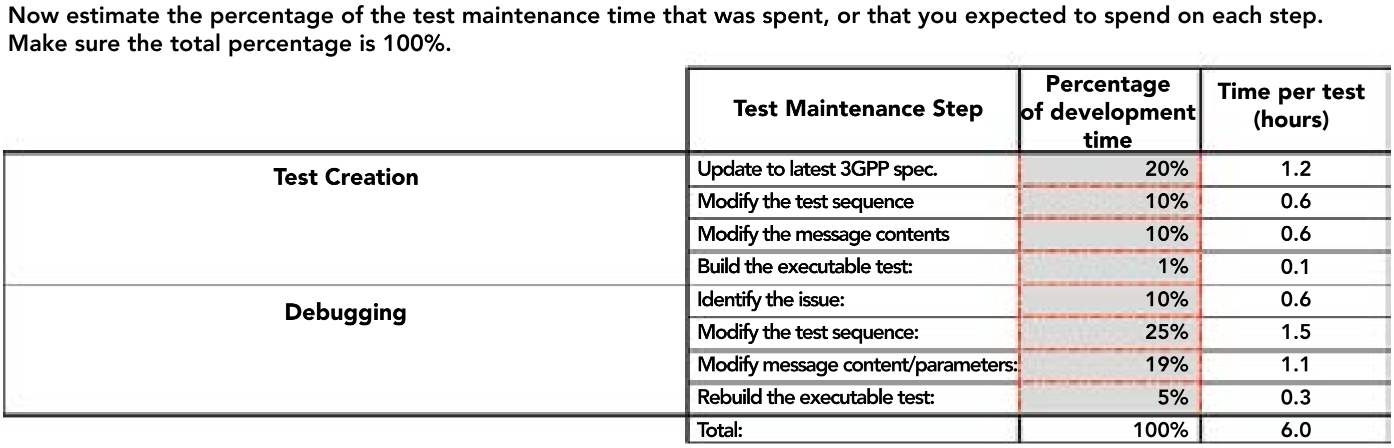


Please enter some assumptions about your recent, or planned test maintenance activity:

Percentage of original test development time required for a maintenance update	50%
Average maintenance time per test:	6 hours
Maintenance duration (months)	3 months
Maintenance cost:	$225,000

Consider your current situation. Let's assume that updating a test for a new stack design, or a new 3GPP release requires 50% of the original test development time. For your suite of 400 tests the maintenance cycle will take 3 months and cost $225k.

Next, we break down the test maintenance activity into tasks and consider the proportion of the total test maintenance time that is spent on each task.

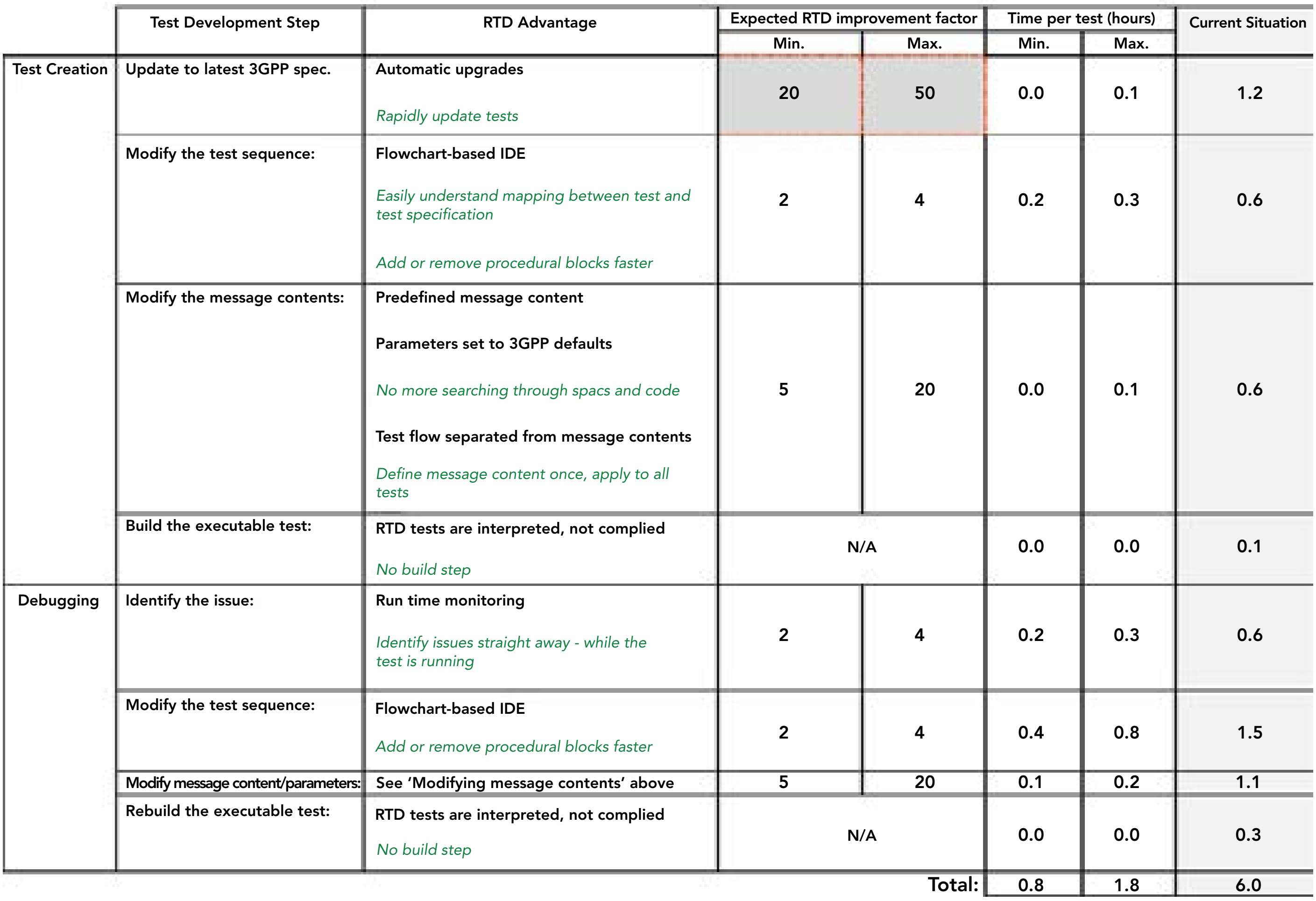


Now estimate the percentage of the test maintenance time that was spent, or that you expected to spend on each step. Make sure the total percentage is 100%.

	Test Maintenance Step	Percentage of development time	Time per test (hours)
Test Creation	Update to latest 3GPP spec.	20%	1.2
	Modify the test sequence	10%	0.6
	Modify the message contents	10%	0.6
	Build the executable test:	1%	0.1
Debugging	Identify the issue:	10%	0.6
	Modify the test sequence:	25%	1.5
	Modify message content/parameters:	19%	1.1
	Rebuild the executable test:	5%	0.3
	Total:	100%	6.0

Now we can consider each task and look at specific RTD features that will reduce the time taken. RTD saves time in your engineer's day to day activities. RTD automatically updates tests to conform to the latest 3GPP specification. As per the development stage, modifying the test sequence and message contents is faster. There is no build overhead and RTD's debugging features enable you to quickly identify any issues.

Now let's see the improvement that RTD will bring:



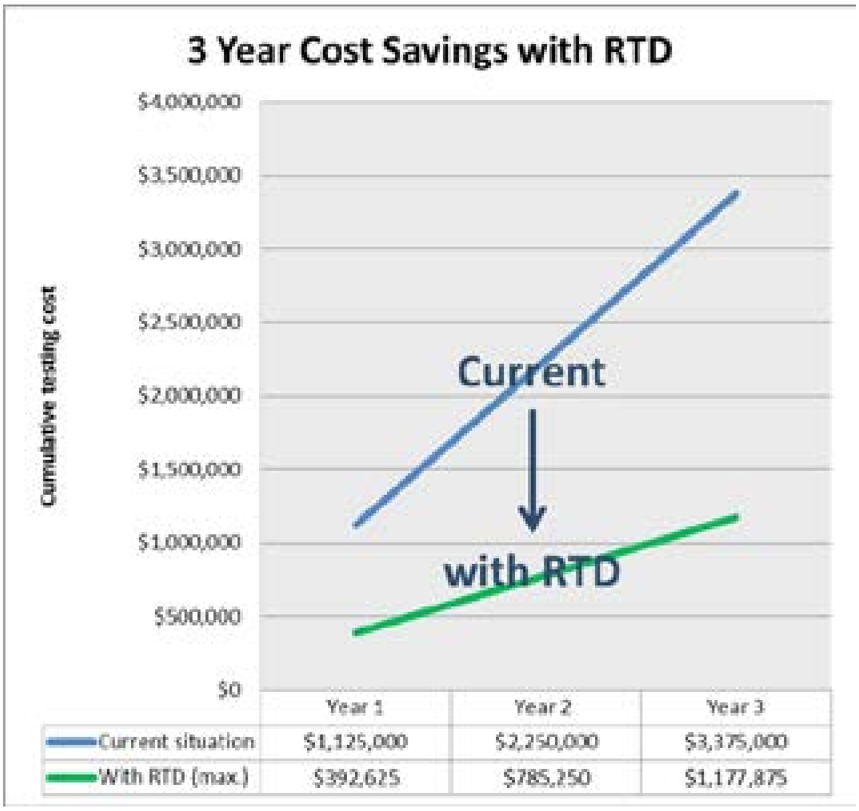
	Test Development Step	RTD Advantage	Expected RTD improvement factor		Time per test (hours)		Current Situation
			Min.	Max.	Min.	Max.	
Test Creation	Update to latest 3GPP spec.	Automatic upgrades *Rapidly update tests*	20	50	0.0	0.1	1.2
	Modify the test sequence:	Flowchart-based IDE *Easily understand mapping between test and test specification* *Add or remove procedural blocks faster*	2	4	0.2	0.3	0.6
	Modify the message contents:	Predefined message content Parameters set to 3GPP defaults *No more searching through spacs and code* Test flow separated from message contents *Define message content once, apply to all tests*	5	20	0.0	0.1	0.6
	Build the executable test:	RTD tests are interpreted, not complied *No build step*	N/A		0.0	0.0	0.1
Debugging	Identify the issue:	Run time monitoring *Identify issues straight away - while the test is running*	2	4	0.2	0.3	0.6
	Modify the test sequence:	Flowchart-based IDE *Add or remove procedural blocks faster*	2	4	0.4	0.8	1.5
	Modify message content/parameters:	See 'Modifying message contents' above	5	20	0.1	0.2	1.1
	Rebuild the executable test:	RTD tests are interpreted, not complied *No build step*	N/A		0.0	0.0	0.3
				Total:	0.8	1.8	6.0

If we factor these improvements into our model we calculate that using RTD **it will typically take between 0.8 and 1.8 hours to update each test, compared to 6 hours using a traditional solution.** This is consistent with our customers' experience.

This means that **each maintenance cycle will take between 0.4 and 0.9 months, instead of your original 3 months.**

The chart below shows the dramatic cost savings achievable with RTD over a 3 year period. It shows the current cumulative testing costs vs. the expected cost range using RTD using the example parameters above, and assuming 1 development project plus 3 maintenance cycles per year. **It shows that over 3 years, RTD could cut your testing costs by 65% to 80%**

!



	Year 1	Year 2	Year 3
Current situation	$1,125,000	$2,250,000	$3,375,000
With RTD (max.)	$392,625	$785,250	$1,177,875

Remember this model only shows the savings that RTD brings about for the day-to-day test development activities. It is one component of the overall exceptional ROI that RTD will deliver to your organization. Overall, RTD will improve development efficiency, reduce capital investment, improve product quality and ease human resourcing. Your business is unique and you know it best. Please use the table on the next page as a guide to help you to estimate the total time and money that RTD will save your organization.

		RTD	
		Money saved	Time saved
Improved Development Efficiency	Faster test creation		
	Faster test log analysis		
	Zero test rebuild time		
	Test automation		
	Faster 3GPP specification upgrades		
	Improved reuse, sharing and management of tests		
	Faster system set up		
Reduced Capital Investment	Built-for-purpose, no adaptation needed		
	No need for special equipment to test newest features		
	Reuse platform for carrier acceptance and conformance test		
	Design and analyze with no system hardware		
	View protocol logs without special software		
Improved Product Quality	Reduced software rework		
	Fewer false pass results		
Easier Human Resourcing	Lower cost staff		
	Lower recruitment costs		
	Faster recruitment		
	Lower training costs		

Summary and Conclusion

Fixing problems early is a business goal for any wireless product development organization and correctly implementing protocols the first time round will reduce cost and improve product quality. In order to meet these goals you will need to hire world-class engineers and equip them with the most effective development tools and test solutions. There are many benefits to be gained by adopting a purpose-built signaling tester as the heart of a successful UE development and testing strategy. The Rapid Test Designer (RTD), coupled with Anritsu's preeminent range of mobile signaling testers has been adopted by the world's leading device makers and network operators as the unsurpassed solution for rapidly testing and debugging wireless mobile device signaling. We have developed a model which allows you to calculate the time to market advantage and operational cost savings that RTD can deliver relative to an alternative solution. For our illustrative example project, RTD gave a time to market advantage in the region of 3 to 4 months with cost savings of over $500k. When we also consider the other benefits such as reduced personnel and training costs, time saved by self-documenting tests, time saved updating test suites to newer standards, easier system setup and configuration and improved test automation, then these significant operational savings only represent a proportion of the total time and money that RTD can save your organization. It is easy to understand why RTD has been adopted by the world's leading mobile protocol developers.

Anritsu

Specifications are subject to change without notice.

● **United States**
Anritsu Company
1155 East Collins Blvd., Suite 100, Richardson,
TX 75081, U.S.A.
Toll Free: 1-800-267-4878
Phone: +1-972-644-1777
Fax: +1-972-671-1877

● **Canada**
Anritsu Electronics Ltd.
700 Silver Seven Road, Suite 120, Kanata,
Ontario K2V 1C3, Canada
Phone: +1-613-591-2003
Fax: +1-613-591-1006

● **Brazil**
Anritsu Eletrônica Ltda.
Praça Amadeu Amaral, 27 - 1 Andar
01327-010 - Bela Vista - São Paulo - SP - Brazil
Phone: +55-11-3283-2511
Fax: +55-11-3288-6940

● **Mexico**
Anritsu Company, S.A. de C.V.
Av. Ejército Nacional No. 579 Piso 9, Col. Granada
11520 México, D.F., México
Phone: +52-55-1101-2370
Fax: +52-55-5254-3147

● **United Kingdom**
Anritsu EMEA Ltd.
200 Capability Green, Luton, Bedfordshire, LU1 3LU, U.K.
Phone: +44-1582-433200
Fax: +44-1582-731303

● **France**
Anritsu S.A.
12 avenue du Québec, Bâtiment Iris 1- Silic 612,
91140 VILLEBON SUR YVETTE, France
Phone: +33-1-60-92-15-50
Fax: +33-1-64-46-10-65

● **Germany**
Anritsu GmbH
Nemetschek Haus, Konrad-Zuse-Platz 1
81829 München, Germany
Phone: +49-89-442308-0
Fax: +49-89-442308-55

● **Italy**
Anritsu S.r.l.
Via Elio Vittorini 129, 00144 Roma, Italy
Phone: +39-6-509-9711
Fax: +39-6-502-2425

● **Sweden**
Anritsu AB
Kistagången 20B, 164 40 KISTA, Sweden
Phone: +46-8-534-707-00
Fax: +46-8-534-707-30

● **Finland**
Anritsu AB
Teknobulevardi 3-5, FI-01530 VANTAA, Finland
Phone: +358-20-741-8100
Fax: +358-20-741-8111

● **Denmark**
Anritsu A/S (Service Assurance)
Anritsu AB (Test & Measurement)
Kay Fiskers Plads 9, 2300 Copenhagen S, Denmark
Phone: +45-7211-2200
Fax: +45-7211-2210

● **Russia**
Anritsu EMEA Ltd.
Representation Office in Russia
Tverskaya str. 16/2, bld. 1, 7th floor.
Russia, 125009, Moscow
Phone: +7-495-363-1694
Fax: +7-495-935-8962

● **United Arab Emirates**
Anritsu EMEA Ltd.
Dubai Liaison Office
P O Box 500413 - Dubai Internet City
Al Thuraya Building, Tower 1, Suit 701, 7th Floor
Dubai, United Arab Emirates
Phone: +971-4-3670352
Fax: +971-4-3688460

● **India**
Anritsu India Private Limited
2nd & 3rd Floor, #837/1, Binnamangla 1st Stage,
Indiranagar, 100ft Road, Bangalore - 560038, India
Phone: +91-80-4058-1300
Fax: +91-80-4058-1301

● **Singapore**
Anritsu Pte. Ltd.
11 Chang Charn Road, #04-01, Shriro House
Singapore 159640
Phone: +65-6282-2400
Fax: +65-6282-2533

● **P.R. China (Shanghai)**
Anritsu (China) Co., Ltd.
Room 2701-2705, Tower A,
New Caohejing International Business Center
No. 391 Gui Ping Road Shanghai, 200233, P.R. China
Phone: +86-21-6237-0898
Fax: +86-21-6237-0899

● **P.R. China (Hong Kong)**
Anritsu Company Ltd.
Unit 1006-7, 10/F., Greenfield Tower, Concordia Plaza,
No. 1 Science Museum Road, Tsim Sha Tsui East,
Kowloon, Hong Kong, P.R. China
Phone: +852-2301-4980
Fax: +852-2301-3545

● **Japan**
Anritsu Corporation
8-5, Tamura-cho, Atsugi-shi, Kanagawa, 243-0016 Japan
Phone: +81-46-296-1221
Fax: +81-46-296-1238

● **Korea**
Anritsu Corporation, Ltd.
502, 5FL H-Square N B/D, 681
Sampyeong-dong, Bundang-gu, Seongnam-si,
Gyeonggi-do, 463-400 Korea
Phone: +82-31-696-7750
Fax: +82-31-696-7751

● **Australia**
Anritsu Pty. Ltd.
Unit 21/270 Ferntree Gully Road, Notting Hill,
Victoria 3168, Australia
Phone: +61-3-9558-8177
Fax: +61-3-9558-8255

● **Taiwan**
Anritsu Company Inc.
7F, No. 316, Sec. 1, NeiHu Rd., Taipei 114, Taiwan
Phone: +886-2-8751-1816
Fax: +886-2-8751-1817

1309

Please Contact:

Printed on Recycled Paper