

# **JMPS**

# **Task Decomposition**

# **Process**

**A Structured Approach to Capturing  
User Requirements**





# Overview

---

- **The Big Picture**
- **Task Decomposition Fundamentals**
- **Your Role in the Process**



# The Big Picture

---

- **What are we trying to do?**
  - Develop a process to capture a comprehensive set of user requirements for JMPS v2 and beyond
  - Adequately and formally document user requirements in order to initiate the systems engineering phase and provide accountability back to the users
  - Achieve validation from the experts of both the process and the captured requirements



# Our Approach

---

- **Use a structured analytical approach to the problem**
  - Establish a formal model of the problem
  - Adhere to a defined set of rules wrt the model
  - Allow for traceability in requirements capture
  - Scrutinize the process at regular intervals for sanity checks (validation, quality assurance, risk mitigation, etc.)

# Why Do It This Way?



- **Need a repeatable way to make sense of the user requirements chaos**
  - You need to be able to see the BIG PICTURE, what's covered, what's not...
- **Need to identify commonality**
  - CNO directive: no more stovepipe systems
  - Limited funding: can only afford to build things once
  - One buzzword: interoperability
- **Requirements must be identified or verified by the users. After all, this Bud's for you...**
- **Don't want to repeat mistakes made in the past**





# TDD Fundamentals

---

- What is it?
  - High level view of ALL TASKS that need to be accomplished in order to conduct mission planning.
  - Model starts with general tasks and decomposes these tasks hierarchically to show greater levels of detail.
  - Diagram is stopped at five levels. This precludes limiting the system design space.
  - A means to the real end product: URdb

# TDD Fundamentals (not)

---



- **What It is not...**
  - The TDD is not a design of JMPS itself.
  - It is not a flowchart. There is absolutely no indication of sequence in this model. The order in which tasks are accomplished varies by user/platform/mission. Not an easy way to identify common tasks...
  - No task is more important than any other. The numbers are there to help us track them

# TDD Rules (say it with me...)



## ***1) We are only interested in USER requirements***

- Don't try to design the system, a whole team of system engineers will do that for you. Focus on identifying WHAT you want, not HOW to build it...

## ***2) The TDD is not a flowchart***

- You may think in sequence to help organize your thoughts, but don't try to illustrate order in the diagram...

## ***3) No task ever repeats anywhere in the diagram***

- Repetition is an immediate indicator that you are trying to show sequence...

## ***4) Try to think in general terms***

- Remember: we are trying to figure out commonality. Also, if you get too specific, you start to limit the design options. Also keep in mind that there are 11 more warfare areas to do this after you...

## ***5) Lower levels are not necessarily 100% decomposition of the upper level***

- A portion of the stated task may be accomplished at the upper level with further granularity of the remaining portion broken out below...

# Your Role in the Process



- **Divide into working groups by platform**
- **Check our work so far**
  - We've already mapped over a dozen valid documents into the TDD. Ask questions, we will show you where any linked need statement came from. Look at where other platforms have identified needs.
- **Use the TDD to grasp the BIG PICTURE**
- **Make your changes to the TDD**
  - Things we missed, things we didn't think of, things you don't think are valid.
  - Remember, the TDD, CONOPS and Urdb are parts of a single integrated process. Change to one = change to all...
- **Give us feedback on the approach**
  - Remember this is a prototype. You are the first warfare area to work the problem. Tell us what made sense, what didn't...



# TDD Rules (say it with me...)

## ***1) We are only interested in USER requirements***

- Don't try to design the system, a whole team of system engineers will do that for you. Focus on identifying WHAT you want, not HOW to build it...

## ***2) The TDD is not a flowchart***

- You may think in sequence to help organize your thoughts, but don't try to illustrate order in the diagram...

## ***3) No task ever repeats anywhere in the diagram***

- Repetition is an immediate indicator that you are trying to show sequence...

## ***4) Try to think in general terms***

- Remember: we are trying to figure out commonality. Also, if you get too specific, you start to limit the design options. Also keep in mind that there are 11 more warfare areas to do this after you...

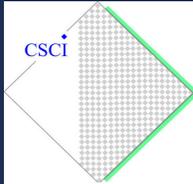
## ***5) Lower levels are not necessarily 100% decomposition of the upper level***

- A portion of the stated task may be accomplished at the upper level with further granularity of the remaining portion broken out below...



# How to Make Changes

- **Every working group is assigned a color. Any change you want to make to the TDD will be done with colored post-it-notes.**
  - Thou shalt not draw on the diagram...
- **Fill out a TDD worksheet for any addition or change your group wants to make.**
  - For every sticky, thou shalt fill out a matching worksheet...
- **Finalize your work by the end of tomorrow, and present your changes to the rest of the group Wednesday morning.**
  - Thou shalt not procrastinate. You have lots more to do before we will let you leave...



# Joint Mission Planning System Task Decomposition Worksheet



Working Group	Suggested TDD#	CONOPS Reference Para.
HS ___ HSL ___ HM ___ VP ___ VS ___ All ___		
<b>Task:</b> _____ _____ _____		
<b>Task Description:</b> _____ _____ _____ _____ _____		
<b>User Requirements Statement:</b> _____ _____ _____ _____ _____		
<b>Notes:</b> _____ _____ _____ _____ _____ _____		
<b>Source Document reference:</b> _____ _____ _____ _____		



# TDD Rules (say it with me...)

## ***1) We are only interested in USER requirements***

- Don't try to design the system, a whole team of system engineers will do that for you. Focus on identifying WHAT you want, not HOW to build it...

## ***2) The TDD is not a flowchart***

- You may think in sequence to help organize your thoughts, but don't try to illustrate order in the diagram...

## ***3) No task ever repeats anywhere in the diagram***

- Repetition is an immediate indicator that you are trying to show sequence...

## ***4) Try to think in general terms***

- Remember: we are trying to figure out commonality. Also, if you get too specific, you start to limit the design options. Also keep in mind that there are 11 more warfare areas to do this after you...

## ***5) Lower levels are not necessarily 100% decomposition of the upper level***

- A portion of the stated task may be accomplished at the upper level with further granularity of the remaining portion broken out below...

# Top Level Description



- **TASKING**
- **RESEARCH AND ANALYSIS**
- **PLAN CONSTRUCTION**
- **COORDINATION, COLLABORATION AND COMMUNICATION**
- **PLAN EXECUTION**
- **POST MISSION REPORTING**
- **MISSION PLANNING ENVIRONMENT**
  - Detailed review will be done by sections, starting tomorrow morning

# Check It Out...



- Copies of the TDD are all around the room.
- Hit the boards and start to absorb the big picture.
- Look at where your community does not have links established
- Think about what you feel is missing...

